1995年10月1日発行（毎月1回1日発行）第14巻10号通巻162号 昭和58年11月2日第三種郵便物認可

PERSONAL COMPUTER MAGAZINE for MZ, X1, and X68000

Oh!X

特集 Now Printing

根性の文字出力/輪郭保護拡大処理/減色処理と印刷/HCESCV2I.X

新刊紹介　SX-WINDOW ver.3.1開発キット

新連載　Lisp一夜漬け/新製品紹介　マルチシンクモニタPC-TM151

10
1995



SOFT BANK オー/エックス
定価760円

SHARP

目の付けどころが、シャープでしょ



■実画面：1,024×1,024ドット、表示画：768×：512ドット

●画面は広告用に作成した、機能を説明するためのイメージ画面です。また、各種アイコンなどは、SX-WINDOW ver.3.1がもつ機能を使って作成したもので、標準装備のものとは異なるものもあります。
●本広告中の「シャーペン」で表示している文字のフォントはツァイト社の、「書体倶楽部」のフォントを使用しています。

❶「パターンエディタ」で作成したデータを背景に設定可能。

❷日本語フロントプロセッサ ASK68K ver.3.0の辞書メンテナンスがウィンドウ上で可能。

❸ESC/Page.LIPSIII.PostScriptに対応したプリンタが利用できます。

❹付属アプリケーション「シャーペン」編集例。文字ごとに文字種・文字の大きさの指定、装飾が可能。またインライン入力をサポート、イメージデータの貼りつけもOK。

❺512×512ドットの範囲内で65,536色の表示が可能。

❻「CGAウィンドウ」、65,536色（最大）のコンピュータアニメーション表示が可能。

❼異なる画像フォーマットへのコンバートが可能。

❽アイコンデータや背景データを作成する「パターンエディタ」。

❾オリジナルに作成したアイコンパターンの例。

❿Human68kやX-BASICのコマンドをSX-WINDOWアプリケーションと同時にタイムシェアリングで実行できます。

■「SX-WINDOW ver3.1システムキット」のお問い合わせは…シャープ㈱電子機器事業本部（液映）システム機器推進プロジェクトチーム　〒162 東京都新宿区市谷八幡町8番地 ☎(03)3260-1161(大代表)へ

フィールドが、膨らむ。

X68030
32bit PERSONAL WORKSTATION
&
X68000
PERSONAL WORKSTATION・XVI

68買ったら
EXEクラブ
へ入ろう！

先が、ますます面白くなる。

●

未来への確かなビジョンをベースに
発展性のあるプラットホームとしてのウィンドウ環境を提供する
国産オリジナルウィンドウシステムSX-WINDOW。

●

GUI環境や操作環境、高速化へのゆるぎない探求、
マルチメディアの統合的なハンドリング。

●

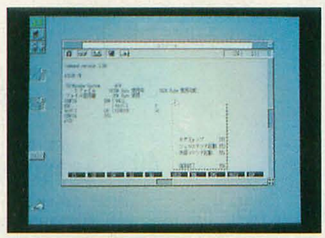
いま、より多彩なフィールドへ
そのインテリジェンスが展開を始める。

●

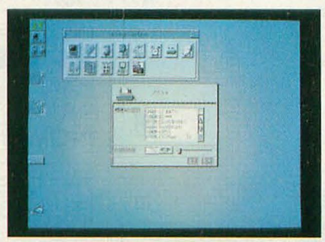
次のステージが見えてくる。



●インライン入力のサポート：ASK68K Ver.3.0を利用したインライン入力をSX-WINDOWで実行可能。またシャーペン.Xをワープロとして利用できるよう、さまざまな機能が付加されています。

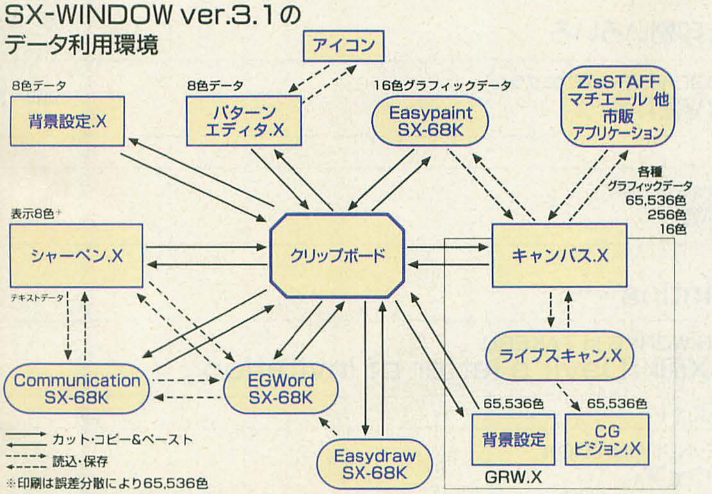


●コンソールをサポート：Human68kやX-BASICのコマンドをSX-WINDOWアプリケーションと同時にタイムシェアリングで実行できます。
（グラフィックを利用したものなど、SX-WINDOWと処理が重複するものは実行できません。）



●多彩なプリンタに対応：さまざまなSX-WINDOWアプリケーションで利用できるページプリンタドライバを標準装備。ESC/Page、LIPS III、PostScriptに対応したプリンタが利用できます。

SX-WINDOW ver.3.1の
データ利用環境



EXE
クラブって
何だ？

X68030/X68000を手に入れて、いろいろチャレンジしたい皆さん。情報のチャンネルは多いほどいいですよね。ということで、EXEクラブは68ユーザーのための水先案内人。あなたのチャレンジを強力にバックアップしますよ。

本体同梱の入会申込ハガキを送るだけで、自動的に無料入会。さらに下記の特典付き。

メリット 1
会員ナンバー入りのオリジナル会員電卓がもらえる。

メリット 2
各種フェアご優待・イベント案内等、数々の特典がある。

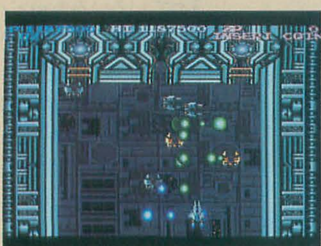
今も、先も楽しめる。
いつも新展開の予感、SX-WINDOWのニューバージョン。

SX-WINDOW ver.3.1

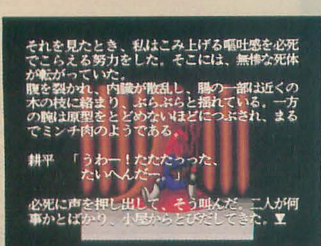
「SX-WINDOW ver3.1システムキット」CZ-296SS（130mmFD）/CZ-296SSC（90mmFD）標準価格22,800円（税別）


特集 Now Printing


ジェノサイド2


沙羅曼蛇


X68000ゲーム年代記

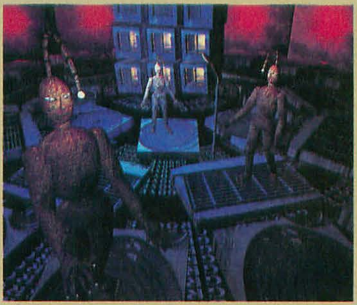


THE USER'S WORKS in TAKERU


マルチシンクモニタPC-TM151

Oh!X CONT

〈スタッフ〉
●編集長／前田 徹 ●副編集長／植木章夫 ●編集／山田純二 高橋恒行 ●協力／有田隆也 中森 章 林 一樹 吉田幸一 華門真人 朝倉祐二 大和 哲 村田敏幸 丹 明彦 三沢和彦 長沢淳博 清瀬栄介 柴田 淳 瀧 康史 横内威至 進藤慶到 菊地 功 伊藤雅彦 ●カメラ／杉山和美 ●イラスト／山田晴久 江口響子 高橋哲史 川原由唯 ●アートディレクター／島村勝頼 ●レイアウト／元木昌子 加藤真二 ●校正／フィールドアップ



表紙絵：塚田　哲也

E　N　T　S

1995 OCT.
10

■広告目次

SHARP

目の付けどころが、シャープでしょ。

1,677万色対応、ビデオ映像を高画質・高速取り込み

テレビやビデオ、ビデオディスクなどの映像をX68シリーズやMacシリーズ※1の動画・静止画データとして高速取り込みが可能、いわば“ビデオスキャナ”とでも呼びたいビデオ入力ユニットです。1,677万色対応、最大640×480ドットの高解像度※2。動画・静止画の手軽なハンドリングが、新たなグラフィックシーンを創造します。

※1 MacintoshはIIシリーズ以降の機種に対応、ディスプレイ解像度が640×480ドットの場合、取り込み可能な範囲は、160×120ドット、320×240ドットのサイズになります。
※2 X68030/X68000シリーズでは、1,677万色はデータ作成のみに対応。表示は最大65,536色、解像度は512×512ドット。また、Macintoshは機種により表示色数が異なります。

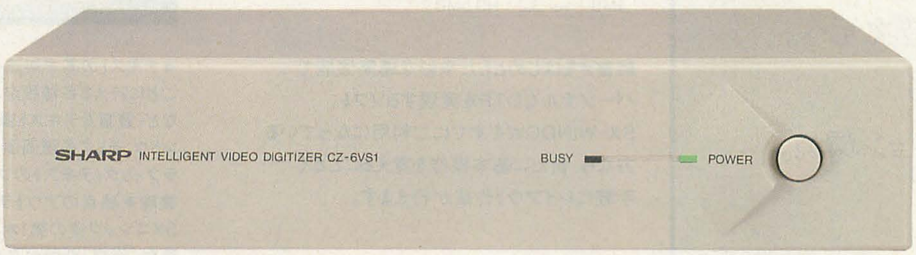
アプリケーションツール「ライブスキャン」を標準装備

動画や静止画を簡単に保存できるアプリケーションソフト「ライブスキャン」※を標準装備。取り込んでいる映像を表示したり、残したいシーンを簡単に静止画保存したり、手軽な動画・静止画ハンドリングでパソコンの可能性をさらに広げます。X68030/X68000シリーズ用SX-WINDOW対応版とMacintoshシリーズ用QuickTime対応版の2種類を同梱しています。



※SX-WINDOW版はバージョン3.0以降（メモリー4MB以上）、QuickTime版はMacintosh漢字Talk7リリース7.1以上のシステムとQuickTime1.5以上（メモリー8MB以上）が必要です。

1,677万色対応の高速映像取り込み、動画・静止画の手軽なハンドリングが、新たなマルチメディアシーンを創造する。



■SCSIインターフェイス採用：パソコンの専用I/Oスロットを使わずに接続可能になり、汎用化を実現しました。またSCSI-2（FAST）インターフェイスの採用により、データ転送速度の高速化を図っています。X68030/X68000シリーズでは、SCSI-2（FAST）対応のハードディスクを接続することにより、パソコン本体を経由しないで、ハードディスクに直接、動画データをテンポラリデータとして記録することが可能です。パソコン本体のハードディスクへは、記録終了後に、テンポラリデータを変換し動画データとして保存できます。

※CZ-600C/601C/611C/602C/612C/652C/662C/603C/613C/653C/663Cに接続する場合は別売のSCSIインターフェイスボードCZ-6BS1ならびにSCSI変換ケーブルCZ-6CS1が必要です。※CZ-604C/623C/634C/644Cに接続する場合は、別売のSCSI変換ケーブルCZ-6CS1が必要です。
※Macintosh Power Bookシリーズに接続する場合は別売のSCSIケーブルなどが必要です。詳しくはMacintosh Power Bookシリーズの取扱説明書をご覧ください。

■高機能MPUを搭載：クロック周波数25MHzの32ビットMPU/MC68EC020を搭載、高速処理やパソコン本体の負担の軽減を実現します。

●MacはMacintoshの略称です。●Macintosh、Macintosh IIは、米国アップルコンピュータ社の登録商標です。●Power Bookは米国アップルコンピュータ社の商標です。●漢字Talk7はアップルコンピュータジャパン社の商標です。●QuickTimeは、米国アップルコンピュータ社の商標です。●価格には、消費税及び配送・設置・付帯工事費、使用済み商品の引き取り費等は含まれておりません。

for X68 Mac

ビデオ入力ユニット

CZ-6VS1

標準価格178,000円（税別）

SHARP

目の付けどころが、
シャープでしょ

For X68030/X68000series

ORIGINAL SOFTWARE COLLECTION

さらに高度な創造次元へ。
ますます成熟する
そのアプリケーション環境。

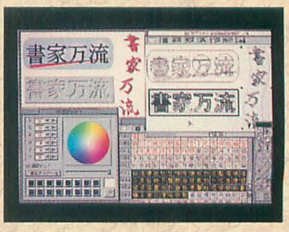
X68030
32bit PERSONAL WORKSTATION



アプリケーション

●独自のアウトラインフォントを付属

フォント&ロゴ デザインツール 書家万流 SX-68K



CZ-282BWD　標準価格29,800円(税別)

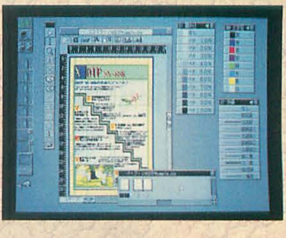
4MB | ver.3.0 | HD 10MB

フォントやロゴを手軽に作成するための
デザインツール。作成したロゴは
クリップボードを介し、シャーペンや
EGWord SX-68K、XDTP SX-68Kなど
他のアプリケーションで利用できます。

●SX明朝体/SXゴシック体フォント(JIS第1水準&第2水準)を付属●ベジェ曲線のアウトライン編集によるデータ作成●フォントファイル全体にわたってのエフェクト処理●既存のフォントファイルからのデータ抽出、ドローオブジェクトへのエフェクト処理●複数のフォントファイルをリンクして新たなフォントファイルの作成が可能●65,536色表示で確認しながらロゴ作成ができるグラフィックウィンドウ(GRW.X)対応

●パーソナルDTPをX68で

XDTP SX-68K



CZ-291BWD　標準価格35,000円(税別)

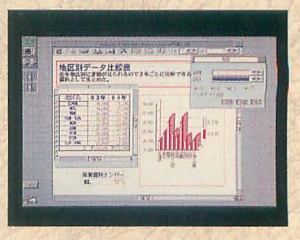
4MB | ver.3.0 | HD 5MB

縦書きをはじめとした多彩な編集機能で
パーソナルなDTPを実現するソフト。
SX-WINDOWをすでにご利用になっている
方なら、新たに基本操作を覚えることなく
手軽にレイアウト作成が行えます。

●テキストの基本処理をはじめ、テキストフレームごとに行える各種設定、スタイル別の検索/置換など、豊富なテキスト編集機能●グラフィックウィンドウ、そして各種画像フォーマットへの対応●グラフィック/テキストのフレームから独立した罫線機能●独自のアウトラインフォント(SX明朝体、SXゴシック体の第1水準)標準添付●ページの移動/作成/削除がスピーディに行える独立したページウィンドウをサポート

●DTP感覚で自在にレイアウト編集

Datacalc SX-68K



CZ-273BWD　標準価格59,800円(税別)

4MB | ver.3.0 | HD 3MB

SX-WINDOW対応の新世代統合ソフト。
表計算、グラフ、データベース、テキスト、罫線の
各データを1枚の用紙に重ね合わせ、
移動、サイズ変更など
DTP感覚でレイアウト編集ができます。

●カルクシートでは、セル番地を意識することのない直感的なセル指定が可能●データベースフィールドでは、同一項目でもデータ型/データ長の異なったデータを管理できるなど、自由な設計が特長●データベースフィールドで入力したデータをカルクシートのデータとして利用したり、カルクシートのデータ変更を自動的にグラフ表示に反映させたり、同一データからさまざまな分析が可能なデータリンクもサポート

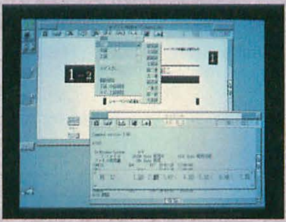
システム & アプリケーション

●さらに実用的なウィンドウシステムへの進化

SX-WINDOW ver3.1 システムキット

CZ-296SS(130mmFD)/CZ-296SSC(90mmFD)　標準価格22,800円(税別)　4MB

ASK68K ver.3.0を利用したインライン入力のサポート、Human68k/BASICコマンドをSX-WINDOWアプリケーションと同時にタイムシェアリングで実行できるコンソールのサポートをはじめ、シャーペン.Xをワープロとして利用できるよう機能アップ。また、さまざまなSX-WINDOWアプリケーションで利用できるページプリンタドライバを標準装備。ドローデータ(FSX)/フォントデータ(IFM)処理の高速化も実現しています。



※コンソールでは、SX-WINDOWと処理が重複するものは実行できません。

●SX-WINDOWを楽しく使うためのアクセサリ集

SX-WINDOW デスクアクセサリ集

CZ-290TWD　標準価格14,800円(税別)

SX-WINDOWをさらに便利に、楽しく使うためのデスクアクセサリ集です。スクリーンセーバ、スクラップブック、アドレス帳、電子手帳通信ツールなど、12種の豊富なアクセサリが収められています。



4MB　ver.3.0

●SX-WINDOW対応ドローイングツール

Easydraw SX-68K

CZ-264GWD　標準価格19,800円(税別)　4MB　ver.3.0

イラスト、フローチャート、地図、見取り図など各種グラフィックが製図感覚で作成できます。作成したデータは他のSX-WINDOW対応アプリケーションでも利用でき、企画書などの作成をサポートします。



●ウィンドウ対応のグラフィックツール

Easypaint SX-68K

CZ-263GWD　標準価格12,800円(税別)　2MB　ver.1.1

マウスによる簡単操作、65,536色中16色の多彩な表現、クリエイティブマインドに応えるウィンドウ対応のペイントツールです。同時に複数のウィンドウを開いて編集でき、各ウィンドウ間のデータ交換も行えます。

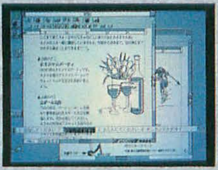


●定評のGUI対応ウィンドウワープロ

EGWord SX-68K

CZ-271BWD　標準価格59,800円(税別)

キャラクタベースのワープロを超えたGUIによる、手軽なDTPソフトとしても優れた表現力を発揮。定評ある日本語入力方式によるインライン入力、各種グラフィックデータやテキストデータの貼り込みができます。
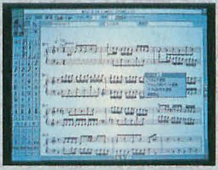


4MB　ver.2.0　HD 5MB

●グラフィック感覚の楽譜入力をサポート

MUSIC SX-68K

CZ-274MWD　標準価格38,000円(税別)

MIDI、FM、ADPCMに対応した楽譜ワープロ&作曲演奏ソフト。自由なレイアウトで、グラフィックを描くように楽譜入力。全パートの同時入力・編集、自動伴奏機能、多彩なプリンタ対応で美しい印刷も行えます。



4MB　ver.3.0

●マルチタスク機能をはじめ通信環境がさらに充実

Communication SX-68K

CZ-272CWD　標準価格19,800円(税別)

通信環境をさらに高めたウィンドウ対応の通信ソフト。マルチタスク機能により他のアプリケーションを実行中でも簡単に通信が可能。自動ログイン機能やプログラム機能など、豊富な機能をサポートしています。



2MB　ver.1.1

開発支援ツール

●X68030/X68000対応開発ツール

C COMPILER PRO-68K ver2.1 NEW KIT

CZ-295LSD　標準価格44,800円(税別)

C compiler PRO-68KのX68030/X68000対応版。従来からの機能に加えて、Human68k ver.3.0、ASK68K ver.3.0にも対応。新たにGPIBライブラリ、MC68882対応フロートライブラリを付属しています。



2MB

●SX-WINDOWソフト開発支援ツール

SX-WINDOW 開発キット Workroom SX-68K

CZ-288LWD　標準価格39,800円(税別)

SX-WINDOW用のソフトウェア開発に必要なツールや33種類のサンプルプログラムを装備。プログラムの編集、リソースの作成、コンパイル、デバッグといった一連の作業がきわめて効率よく実行できます。

※ご使用に当ってはC compiler PRO-68K ver.2.1が必要です。



4MB　ver.2.0

●SX-WINDOW開発キットのサポートツール

開発キット用ツール集

CZ-289TWD　標準価格12,800円(税別)

「SX-WINDOW開発キット」をさらに使いやすくするためのサポートツール集。SXコールの簡易リファレンスを収めたインサイドSX、イベントハンドラ、ヒープビューアなど11種類のツールが用意されています。



4MB　ver.2.0

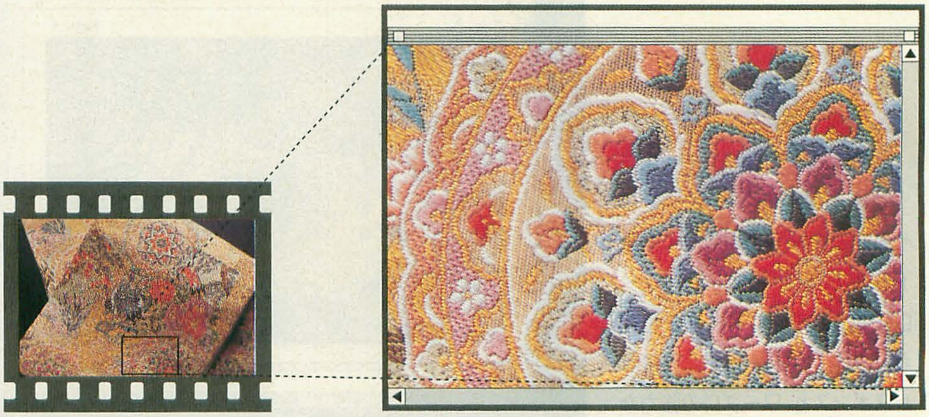
4MB　ver.3.0　HD 10MB の表示は、メインメモリ4MB以上、SX-WINDOW ver.3.0以上、10MB以上の空きのあるハードディスクが必要であることを示しています。●EGWordは株式会社エルゴソフトの登録商標です。

●お問い合わせは…シャープ(株)電子機器事業本部(液映)システム機器推進プロジェクトチーム　〒162 東京都新宿区市谷八幡町8番地　☎(03)3260-1161(大代表)へ　シャープ株式会社

資料請求券 CZソフト Oh!X 10係


高速・高画質、より深まる。
高速・高画質で人気のJX-330がさらに使いやすく! パワーユーザーも納得する実力を実現しました。
2400dpi※1
※1 2400dpiは当社独自手法による疑似解像度です。
＊イメージ写真です。
X68000対応カラーイメージスキャナ
JX-330X
SHARP IS COLOR

SHARP

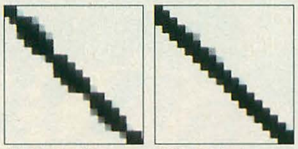


高スピード&高画質により、効率の良い作業を実現。拡大しても画像の荒れが少なく、レタッチ作業の短縮が図れます。
＊画面はハメコミ合成です。

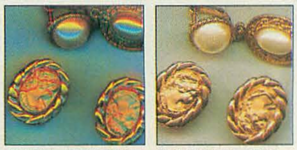
最高2400dpi※1の高解像度を達成。

基本600dpi、最高2400dpi※1の高解像度読み取りで、微細な線や点まで忠実に鮮明に再現します。縮小・拡大は30～2400dpiの範囲で設定可能です。また、約1677万色で原画に忠実なリアルな色合いを再現します。

●シャープ独自の「デジタルズーム機能」により、微細な線やズーム画像も忠実に再現。また、「ワンウェイスキャン方式」を採用し、凹凸のある原稿も鮮明に読み取りできます。



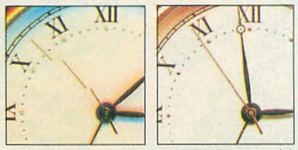
通常の拡大時（当社従来機 JX-325）　デジタルズーム（JX-330シリーズ）



色の付いた影が出る（当社従来機 JX-325）　凹凸物も美しく再現（JX-330シリーズ）

クラス最速※2の高速読み取りを実現。

高速ヘッドリターン（約1秒）と高速読み取りを実現。A4、300dpiならカラー約13秒※3、モノクロなら約1秒※3で読み取りできます。最大A4/リーガルサイズ（216.4×355.6mm）までの原稿の読み取りが可能です。

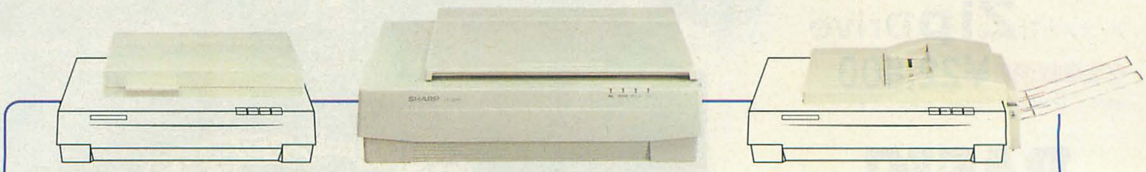


読み取り速度16ms/ラインのスキャナ　読み取り速度3.7ms※4/ライン（JX-330シリーズ）

透過原稿読み取りユニットとADFが同時装着可能。（オプション）

基本解像度600dpiまたは1200dpiの2種類の透過原稿読み取りユニットが選択使用できます。また、最大50枚までの同一サイズの原稿をスピーディーに自動送りできるADFも同時装着できます。





透過原稿読み取りユニット（オプション）
JX-3F6 標準価格 98,000円（税別）
JX-3F12 標準価格 138,000円（税別）

カラーイメージスキャナ
JX-330X 標準価格 178,000円（税別）

ADF［原稿自動送り装置］（オプション）
JX-AF3 標準価格 58,000円（税別）

使いやすい高機能画像入力ソフトを標準装備〈JX-330X〉　●Scanner Tool/s（画像入力ソフト）、対応フォーマット形式：ZIM,PIX,GL3,PIC,GLX,GLM

※1 2400dpiは当社独自手法による疑似解像度です。※2 クラスとは、A4フラットベットクラスのこと。'95年7月現在。※3 室温時（25℃）読み取り開始から読み取り終了までの動作時間。但し、初期動作及びデータ転送時間を除く。※4 室温25℃時。

■消費税及び配送・設置・付帯工事費・使用済み商品の引き取り費等は、標準価格には含まれておりません。

■資料のご請求・お問い合わせは　シャープ株式会社 プリントシステム事業本部プリントシステム営業部
〒545 大阪市阿倍野区長池町22番22号 TEL.(06)621-1221（大代表） FAX.(06)629-1207
〒162 東京都新宿区市谷八幡町8番地 TEL.(03)3267-4410（ダイヤルイン） FAX.(03)3260-2159

シャープ株式会社

SEGA セガサターンマガジン SOFT BANK

SEGA SATURN MAGAZINE

NEXT GENERATION SEGAGAME MAGAZINE

540YEN

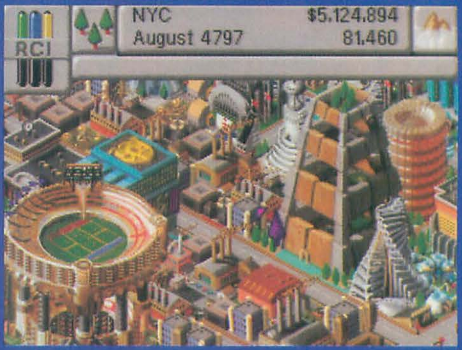
10月号

好評発売中!!
毎月8日発売

特報！

バーチャファイター2
セガラリー・チャンピオンシップ

ストリートファイターZERO／ダークセイバー／ドラゴン・フォース
闘神伝S／スーパーリアル麻雀グラフティ／結婚～Marriage～
ファラドゥーン／天地無用／X-MEN／バーチャレーシング サターン



特集
秋の夜長は都市づくりに励め
建てる!にはまるシミュレーション
サターン版シムシティ2000を徹底紹介、そしてザ・タワーとテーマパークの全貌を速報！

AM2研EXPRESS NEO
短期連載開始!
「VF3」への道／鈴木裕インタビュー

特別企画
最新!サターン周辺機器事情

▼SEGA SATURN PRESS
最新のサターンタイトルをキャッチUP！
X-JAPAN／遥かなるオーガスタ3
機動戦士ガンダム／DOOMII
ドラゴンボールZ
アメリカ横断ウルトラクイズ

▼COMMING SOON SOFT
発売直前！セガサターンソフトを大紹介！
出たなツインビーヤッホー！／QUOVADIS
ガーディアンヒーローズ／ダライアス外伝
レイヤーセクション／宝魔ハンターライム
ワールドアドバンスド大戦略
真・女神転生 デビルサマナー



★★★
特別付録2大ポスター
AM2研特製
バーチャファイター2
宝魔ハンターライム
★★★

■定価は税込みです ■お近くの書店でお求め下さい
ソフトバンク株式会社／出版事業部 販売局 TEL.03-5642-8100

［第53回］
路地売りの花屋

江口　響子

響子inCGわ〜るど

7月末に引っ越しをした。

今年の夏は暑かった。なんでも，気象庁始まって以来，100数年ぶりの暑さなんだそうである。よりによってなんでこんな暑いさなかに……と周囲にいわれつつ，なんとか一段落した。気の遠くなるような荷物の山もほとんど片づいた。

X68000は，ハードディスクこそ壊れなかったものの，電源を入れてみると調子がどうもおかしい。使用可能メモリが異常に少なくなっている。いよいよ寿命かなと心配したが，増設のメモリボードがゆるんでいただけのことだった。こうして，マシンたちも落ち着くべきところに落ち着いた。

今度の住まいには，15畳のリビングルームがある。そこに大きな白いテーブルを置いた。テーブルの上には，いつも季節の花を絶やさないようにしようと決めた。

ご近所の口こみで聞いたおすすめの花屋は，神社の前の路地売りだった。月水金は切り花，火木は鉢植を売る。さっそく，切り花の日に出かけていった。

花屋は，20代後半か30代前半の，よく日に焼けたお兄ちゃんである。お客さんが次から次へとひっきりなしだ。なかには車で花を買いにくる人もいる。

商売はいたってシンプル。生きのいい花を，できるだけ安く仕入れて安く売るのがモットーだそうだ。なるほど元気いっぱいの花たちが，かんかん照りにもめげず，顔を上に向けてすっくと背を伸ばしている。さて，なにを買おうかな〜。

「どれにする？　でも夏は花の種類が少ないんだよね。りんどう，ゆり，白や黄色の菊，トルコききょう，ひまわり……たくさん仕入れてもこの暑さじゃあ〜ね！　花だってかわいそうだよな。え？売れ残ったらどうするかって？　そんときゃ，持って帰って部屋に飾るのさ」

お兄ちゃんは優しいのだった。

「なんてったって信用だよ。少しでもしおれた花を売ったらおしまいだから。こちとら店も構えていない路地売りの花屋だからね。毎日が真剣勝負だよ」





うんうんそのとおり！　肩書きだけで仕事をしている人たちに聞かせたいなあ～。

この間は沖縄の蓮の花を買った。名前は忘れてしまった。「日持ちがいいよ～」とすすめてくれた花は，10日たってもまだ咲いている。

コンピュータを使って仕事をしていると，どうしても部屋にこもりがちだ。季節の移ろいを肌で感じる機会が少なくなる。そんな中で，生花を飾ると，自然のよい香りが部屋に満ちる。一緒に仕事をしているスタッフたちにとっても，みずみずしい花は気分がいいものらしい。

そして季節は秋。これからどんな花を生けようか。

今回のCGデータ

1280×1024ピクセル
1670万色フルカラーを4×5ポジで出力
作成手順
使用ソフトはマチエール。大理石スキャン画像，スケッチ画像取り込み，ネガティブ，カラーレリーフ，指先ツールなどで画像処理。画像処理の部分だけRGB更新セーブ。

暑中見舞いだ!

Oh!X reader'sぎゃらりぃ

気象庁の年頭の長期予想を覆す，昨年に続く猛暑を皆さんいかがお過ごしでしたでしょうか？　編集室は風邪を引きそうなくらいに寒く，酷使されるマシンにとっては快適な夏でした。それでは皆さんから送られてきた暑中見舞いのハガキをどーんと紹介しましょう。



▲岡村　直也(兵庫県)



▲illustration：Y.Kawahara



▲加藤　隆(佐賀県)



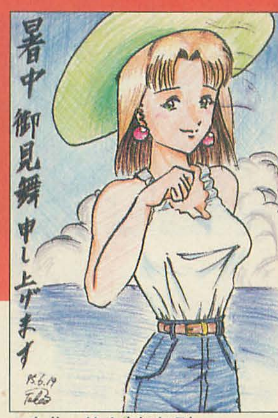
▲清家　亜紀(福岡県)



▲松本　祥子(熊本県)



▲illustration：T.Takahashi



▲今井　健生(奈良県)



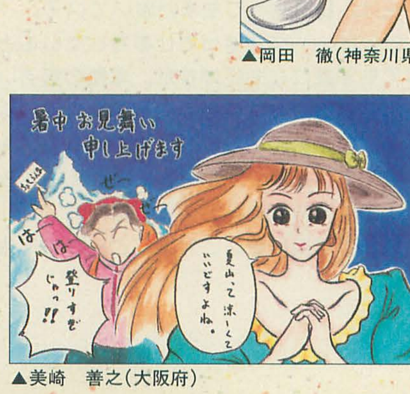
▲岡田　徹(神奈川県)



▲岩瀬　貴代美(福岡県)



▲占部　哲彦(広島県)



▲美崎　善之(大阪府)

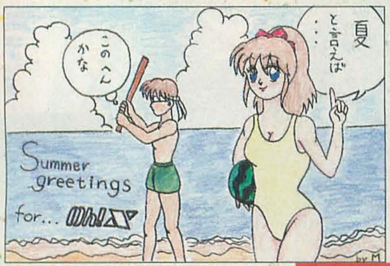


▲豊田　巨（東京都）



▲大高　孝平（宮城県）


▲illustration : Y.Kawahara



▲小田島　倫也（愛知県）



▲佐藤　一秀（愛知県）



▲藤原　彰人（岡山県）



▲青木　一師（奈良県）



▲徳物　信生（広島県）



▲上村　こうじ（広島県）



▲板垣　修（千葉県）



▲武田　正道（兵庫県）

[特集]

Now Printing.....

高性能なカラープリンタが安価になって，グラフィック出力は身近になってきた
ここでは印刷のための画像処理というものを見てみよう



この画像を拡大/減色する



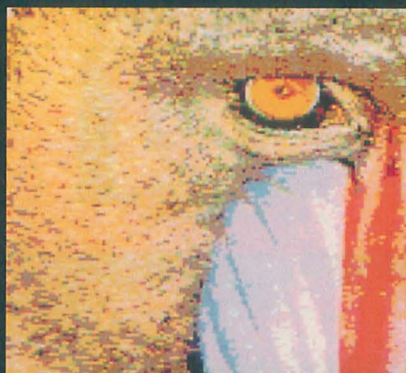
8色で濃度パターン法



ディザリング処理



単純閾値法



16色で濃度パターン法



原画像



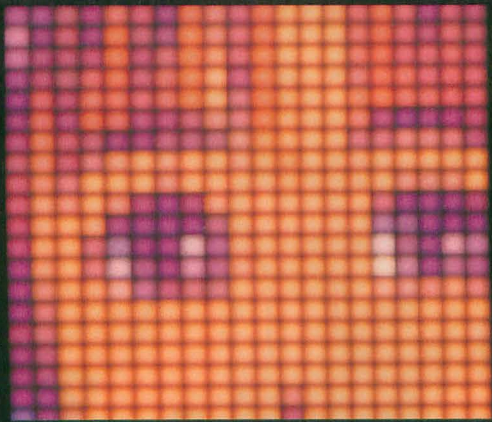
オーダードディザ法



単純閾値法



桒野式拡散法



ガウシャンパルスによる拡大処理



ゼロ補間拡大。ほぼ元画像のまま



線形補間拡大。ぼかしがかかっている



輪郭保存線形補間拡大。ややぼかしが入る



輪郭補正線形補間拡大。積極的に斜め線を補正する



ワープロパックver.2.0での出力（ディザ）



同じく誤差拡散法

同じく粂野式誤差拡散法





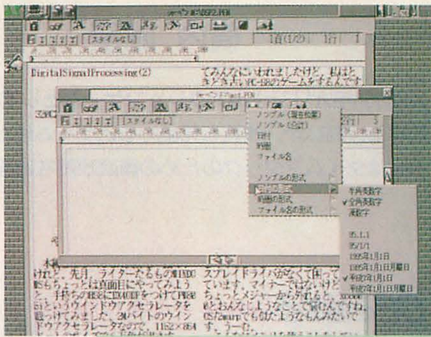
これがESPER MACHだ（原寸）。情報量が多い絵で最近のプリンタの底力を見る。原画は都築和彦氏のCD-ROM「ILUSION」から（部分）

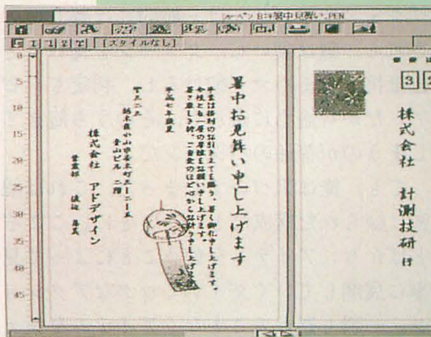


HCESCV21のMACH JETでの出力例（原寸）。上と比べるとややドットが大きいことがわかる

SOFTWARE INFORMATION

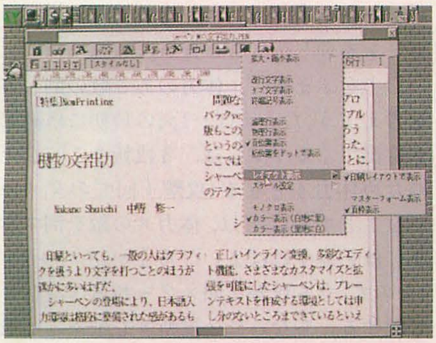
▶シャーペンワープロパックver.2.0





ちょっと製品版のレビューは間に合わなかったが，ここで「シャーペンワープロパックver.2.0」の簡単な概要と開発中の画面写真を紹介しよう。

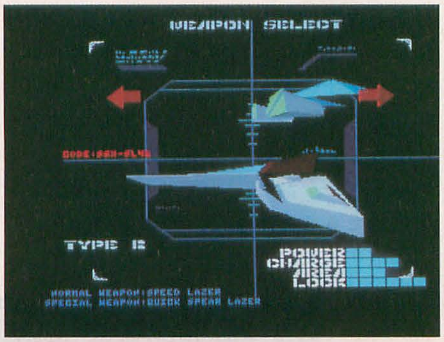
まず，ESC/P V.2とキヤノンBJシリーズ用の専用ドライバが付属となり，SX-WINDOWでの印刷機能が拡張された。両ドライバともラスターグラフィックスコマンド



に対応しており，高精細かつスピーディな印刷が可能だ。さらに，色変換モジュールも拡張され，画像データの表示速度が向上している。なお印刷結果は，特集のカラーページを見ていただきたい。プリンタの性能どおりのハイクオリティな印刷が，SX-WINDOW上で実現されているのがおわかりいただけるだろう。もちろん，印刷機能だけでなく編集機能の強化もされている。新たにレイアウトモードを採用し，印刷時のイメージそのままで編集作業をすることができるようになった。このほかにもマスターフォーム，ノンブル，日付，時刻の印刷，袋とじをサポート。シャーペンだけでなく，SX-WINDOWを強化したい人にもお勧めできるソフトだろう。

X68000用　5″2HD版　9,800円(税別)
計測技研　☎0286(22)9811

▶SION IV





SLASH用のサンプルとして登場してはや2年。今回，途中経過の報告とともに画面写真を公開しよう。ゲームは，前作同様3Dシューティングだが，多少システムに改良が加えられている。まず，ゲームスタート時にオプションを装備にすることによって，自機の使用できる武器が選択できる(3種類)。いままでの回転レーザーとエネルギー弾のほかに，高速レーザー＋槍レーザー，ロングレーザー＋多間節レーザーが使用可能だ。しかも特殊兵器は使用無制限(エネルギーチャージ時間あり)。ガンガンにロックしまくれるぞ。

そして，ゲームは4つの宇宙面と1つの要塞面で構成される予定。敵キャラクターの充実，ステージバリエーションの増加と前作を遥かに凌ぐ作品となりそうだ。

現在のところ，宇宙面のデータがだいたい揃って要塞面のデータ作成に取り掛かっているところ。12月号にはなんとか間に合いそうな気配。期待して待っていてくれ。











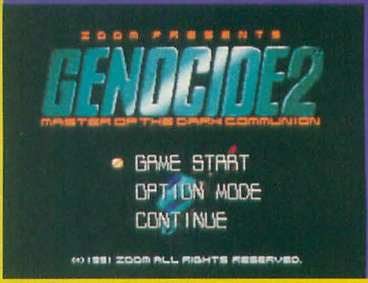
発売中のソフト

★シャーペンワープロパックver.2.0　計測技研　9/11
X68000用　5″2HD版　9,800円(税込)

新作情報

★EXCITINGみるく　TAKERU　10/未
X68000用　5″/3.5″2HD版　1,500円(税込)
★X CASE　Beシステム
X68000用　5″2HD版　19,800円(税込)
★Traum　象スタジオ
X68000用　5″2HD版　価格未定
★麻雀悟空・天竺への道　シャノアール
X68000用　5″2HD版　9,800円(税別)
★地球防衛MIRACLE FORCE　カスタム
X68000用　5″2HD版　価格未定
★プリンセスメーカー　ニュー
X68000用　5″2HD版　14,800円(税別)

●ジェノサイド2



ズーム　☎011(613)0191

秒刻みの華麗なる戦闘（前編）

Yokouchi Takeshi 横内 威至

X68000用オリジナルアクションゲームとして最高峰の「ジェノサイド2」
難易度は高いが、本質を見極めるとゲーム性ががらりと変わるゲームである
今回はタイムアタックのための徹底攻略をお届けする

G2を極める

「ジェノサイド2」は一般的には難しいとされる。敵は硬いし，ダメージを受けると無抵抗のままハマり続けるし，判定もシビア。だから適当に遊んで，そのうち飽きてしまうのが普通のパターンだ。

でも，俺は気づいてしまった。これは綿密に練られた構成だということに。ここぞとばかりにアイテムを使うことによって見事に展開していくダイナミックなアクション。一瞬も遅れることなくアイテムを使えば，見事に次の展開に結びつくのだ。RISCの機械語のように，パイプライン（ベティーの補充だけかな）を考えて行動するのが正しい攻略となる。いかに効率を上げるかを考えるのだ。タイムアタックこそがこのゲームの正しいプレイ方法だ。ちなみに，予想最高タイムは17分30秒前後ではないだろうか（俺の最高タイムは18分03秒）。

つまり，このゲームはアクションというよりもパズルゲームなのである。

操作方法と必殺技

操作は8方向＋2ボタン。ボタンは攻撃とジャンプ。同時押しでアイテムを選ぶ。アイテム選択中もタイムがカウントされることを忘れるな。そして，マニュアルにもない操作テクニックを以下に記す。

1）　高速移動落下

落下中に，斜め上に入力するとダッシュ並の横移動スピードを出すことができる。

2）　ベティー中段攻撃

これは，しゃがんだ状態でのベティーの高さで水平に飛ばす攻撃のこと。しゃがみながら攻撃ボタンを押しておき，立つ瞬間にレバーを横に入れボタンを離せばいい。

3）　ベティーキャンセル

必要もないのにベティーを出しっぱなしにすることはない。余裕があればさっさとエネルギーを溜め，一度ベティーを消して次に備えること。また，使用中のベティーは，ほかのアイテムを選択すると，ゲージをそこから溜め始めることができる。

このほか，ハメ斬り，高速3段斬りといったテクニックもある。これらは，下のカコミを参照してもらいたい。

以上をふまえて，体力のある敵の倒し方を考えよう。たとえば1-1面の最初に格納庫から出てくるヤツらは，3段斬り1回で死ぬため体力4となる（攻撃1回でのダメージを1とする）。では，体力8の敵を倒すときにはどうすればよいだろうか。8＝2＋2＋4なので，立ち攻撃2回×2＋高速3段斬りで殺せる。ちなみに3段斬りで敵はふっ飛ぶが，2機以上だとどれか1機が飛ぶだけでずっと3段攻撃でハメられる。

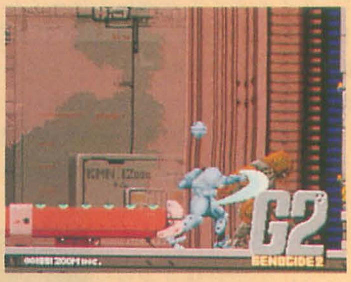
あとは画面モードを15kHzにしておくこと。画面書き換えの周期が31kHzモードよりも若干速いので，タイムに影響が出てくる（長い面では2,3秒は変わってくる）。当然処理落ちしそうなMIDIも避けるべきだ。

それでは各面の攻略を紹介していこう。

テク1　ハメ斬り

通常，攻撃を連射すると3発目で通常の2倍のダメージを敵に与えられるが，敵がふっ飛んでしまい，連続的にダメージを与えられない。そこで，2発目でいったん攻撃を止め，一瞬間をおいてから攻撃を開始すると永遠に攻撃を続けられるのだ。これは，2発攻撃のあとの敵の無敵時間回復よりも，再攻撃のほうが速いためである（しゃがみ攻撃では不可能）。





テク2　高速3段斬り

このゲームでは，普通に立って攻撃するよりもしゃがんで攻撃するほうが速い。ならば，立ち3段攻撃を使うよりも，しゃがみ攻撃を使えばいいように思えるが，しゃがみ攻撃では3段目のダメージが2倍にならないのだ。よって，最初の2発はしゃがみを使い，最後の1発を立ち攻撃することによって，すばやくダメージを与えることが可能となる。





STAGE▸1

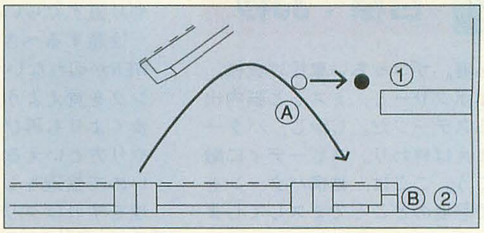
1-1　目標：26秒

まず速攻でベティーを使い，すぐに右に走りつつ飛んでる爆走バイク集団を潰し，格納庫右側で待機。格納庫から出てくる最初の2機を高速3段斬りで殺し，最後の1機をベティーで殺す。次に右に走り，空中に浮かぶ砲台へベティーを当てた直後にエネルギーを溜め始める。そして，右からくるミジンコ(?)をジャンプで越えて右側からMAXベティーを当てる。そのままミジンコ2機をハメ斬り＋ベティーで攻撃。片方が死んだらハメ斬りを使いつつMAXベティーでとどめを刺せ。そして，最後のコンテナの上段にあるEXPLODEを，走りながらベティーを当て続けて壊す。壊れたら右へ行き，格納庫のセンサーを破壊し，クリア直前にEXPLODEを手に入れて終了。ミジンコの行動以外は注意するところはない。



1-2　目標：35秒

ステージ開始後，左上にあるHI-POWERを取って下に戻る。最初のベルトコンベアに乗る前にベティーを出し，コンテナ上にあるセンサー砲台を真下の位置でレバー上を入れて破壊。次の人間コンテナを画面外にあるうちから，ジャンプしてベティーを飛ばして殺す。工作ロボットがすぐ左に待機しているので，高速3段のあと，先に進みながらベティーで潰す。ゲートを越え，中段のコンベアを走り，右に高速移動落下。工作ロボットの右側の人間コンテナの上に乗りベティーで破壊し，工作ロボットを料理しよう。工作ロボットは体力8なので2＋2＋4のハメ斬り×2＋高速3段斬りで殺す。そして人間コンテナを無視して右に走り，右下がりのベルトコンベアに大ジャンプで乗る。コンベアからの落下中にベティーエネルギーを溜め始め，最後の工作ロボットを出現と同時に潰す。最後の場所はセンサーだけをベティーで潰し，ミジンコに乗って砲台の上に乗れば終了。ミジンコはミジンコの中央から見て自機の位置によって方向転換をすることを覚えておこう。



◀①をAの位置で放ったベティーで潰し，Bの位置で②の工作ロボットを叩きのめせ！



▲ジャンプ後のベティーを離す場所を間違うな



▲このベルトコンベアはジャンプ＋レバー上でOKだ

1-3　目標：1分16秒

まずはダッシュして，高速移動落下を使いながら壁挟みの場所へ急ごう。右側の浮遊ブロックに乗って待つ。壁が移動し始めたら大ジャンプで右壁を飛び越え，最上部だけを速攻で破壊する。最上部が壊れればその壁は全滅することを覚えよう。右壁を破壊したあとは，左壁が迫ってくるのをそのまま待ち，左壁もさっさと破壊して上へ戻る。次に現れるミサイル野郎は，相手をせずにブロック上を渡り，MAD-BETTYを取ろう。MAD-BETTYを取ったら高速移動落下を駆使して鉄屑地帯へ直行だ。

そして，回転鉄屑が現れると同時にベティーを出す。これは，上から降ってくる鉄屑を防御する役目を果たしている。次に回転鉄屑へMAXベティーを1発食らわし，さらにベティーのエネルギーを溜める。しばらく待っていると，回転鉄屑野郎本体に当たり判定がつくので，すかさずハメ斬りで右に運び，写真右下のようにハメ殺そう。当然攻撃している間もベティーはずっとエネルギーを溜めておくこと。回転鉄屑野郎を倒すと同時に，あるいはそれ以降すぐにベティーが消えればOKだ。このタイミングでベティーが消えると，ボスにたどり着くまでにベティーが使用可能状態になるのだ。あとは大ジャンプ4回で上まで上がり，ボスへと向かうエレベーターに乗ればいい。





▲右側に潜り込んで，最上部のみを破壊しよう



▲上から降ってくる鉄屑をベティーで防ぎつつ，ハメ殺せ

STAGE1 BOSS

ボスが画面に現れる手前でベティーを出し，気合を込めてボスの上に乗る(ベティーは溜め始めること)。すると，ボスは起き上がって奇声と共に蠢くので垂直ジャンプでかわす。このあと，突起物を出したら真上からMAXベティー＋高速3段斬りを浴びせまくれ。以降これの繰り返しだ。たまに正面を向いて左右に6発屁をこくが，これは垂直ジャンプでかわしてやろう。どうなるかは運次第だ。



▲弱点にMAXベティーを当てろ



▲攻撃は垂直ジャンプでかわせ

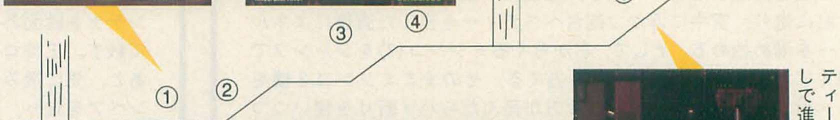
STAGE▸2

2-1 目標：35秒

ステージ開始後のピンク色のヒルは，位置をしっかり覚え，いきなり高速3段斬りで殺す。すぐ右にMAD-BETTYが置いてあるので，手持ちのMAD-BETTYを使ってから取る。そして，そのままダッシュして突っ走る。密集している浮遊爆雷は，はじかれそうになったらすかさず防御すれば勝手に死んでくれるので楽勝だ。そのまま再びヒル，ゲートを破壊し先へ急げ。ここからしばらくは右の図のように進もう。ベティーのエネルギーの溜まるタイミングを見切って突っ走れ！







最初に①を飛び越え，②を右側からしゃがみ3段斬り。②がふっ飛んだら左に追いかけ，MAD-BETTYを使い①，②を破壊（③の浮遊爆雷まで破壊するのが理想）。すぐにベティーを使い，エネルギーを溜めて走り抜け，⑤，⑥を出現と同時に破壊する。追いかけてきた④をぶち壊して，ベティーを溜めつつ先へ急げ



◀このあたりからベティーを溜めっぱなしで進もう

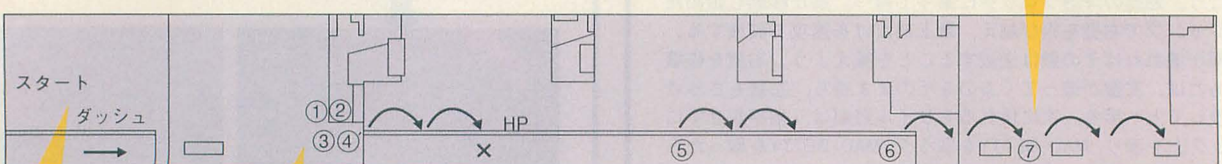
2-2 目標：33秒

いやらしい忍者，ザコっちい車投げ機械，クソ腹立つヘボボクサーと，ミスると脳内出血が盛んになるステージだ。しかし，パターンにハメてしまえば終わり。スピーディに敵を破壊していこう。ここは，結構パターンを作りやすいので忍者のところでミスしてしまいパターンが途切れたら，自決して最初からやり直すぐらいの男気が必要だ。

注意するべきところは特にないが，HI-POWERが切れないうちに忍者を始末するタイミングを覚えよう。また，最後のボクサーは，歩くよりも再び溜め直して当てたほうが賢いやり方といえるが，空中バイクがうっとおしいので無理をして倒す必要はないだろう（無理をすればクリアまで35秒以内も可能か？）。



◀まず、6の忍者をハメ斬りしながら進み、次の7もまとめてハメ斬りにする



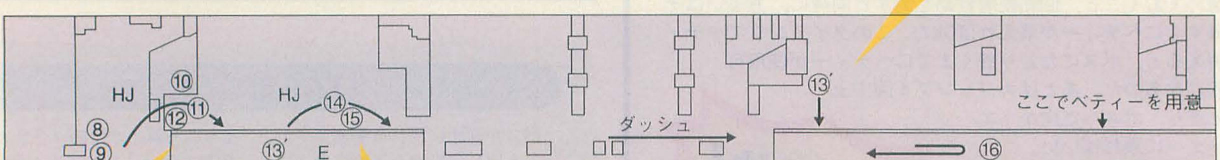




◀まずはHI-POWERを準備し，最初の忍者③を斬り始める。HI-POWER中ならば忍者は，3回斬ると破壊される。なお，忍者は1発ごとに後方へふっ飛ぶので，斬って追いかけてを繰り返して仕留める。次の忍者⑤を右に追いかけて殺すと，すぐに左から追っかけてきた忍者④'と遭遇する。さっさと破壊して先へ進もう。なお，図中のHPはHI-POWERアイテム，Eはエネルギーアイテム，HJはハイジャンプとなっている。



◀このあたりに密集している飛んでいるバイク野郎は，ダッシュ＋ハイジャンプ斬りで，斬り残しのないようにまとめて破壊しろ





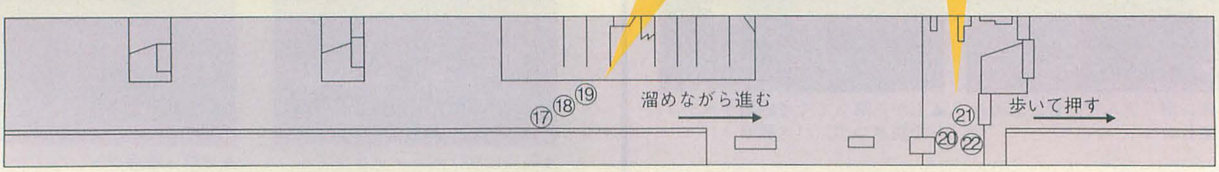


◀⑯の車投げロボットを破壊したら，振り返って，後ろから迫ってくる⑬'の車投げロボットを迎え撃つ。だいたいここでHI-POWERが切れる





◀⑰のボクサーまでにベティーを溜め⑱，⑲を一掃。そのあと，歩きながらベティーでボクサーを押し続け破壊。⑳も同様にして倒す



2-3 目標：1分27秒



▲⑪をサクッとベティーで処理したら，⑫，⑬の工作ロボットを破壊していこう。なお，⑬の工作ロボットは動き始めているのでうかつに突っ込まないように

ここではミスがそのまま30秒のムダとなってしまうのでジャンプ＋ベティー飛ばしと，ハメ斬り，高速3段斬りを駆使して気合を入れて進むべし。全体のマップは下にある図のとおりだ。図の記号は，HJ:ハイジャンプ，SJ:ショートジャンプ，○→:ベティーを放つ位置，E:エレベーターとなっている。

注意すべきは⑩の飛んでるヤツ。こいつは途中の島に降りたと同時にベティーで弾を消し，すぐレバー上で殺す。ミスして下に落ちてしまうと多大なロスとなっていまい，一気にやる気が失せる。最後のフロアは“E-10”と壁に書いてあるあたりから大ジャンプ＋ベティー飛ばしで⑪を破壊せよ。なお工作ロボットは，体力8なので2＋2(ハメ斬り)＋4(高速3段斬り)で破壊せよ。

このステージは敵を倒すタイミングをしっかり覚えていなければ難しい。が，このようなベティーの飼いならしがいかにも「ジェノサイド2」らしいともいえる。ラストフロアの飛んでるヤツを殺して1秒ほどでベティーが消えることになるであろう。ベティー，素敵だ。まさに男の中の男，男玉と呼べよう。



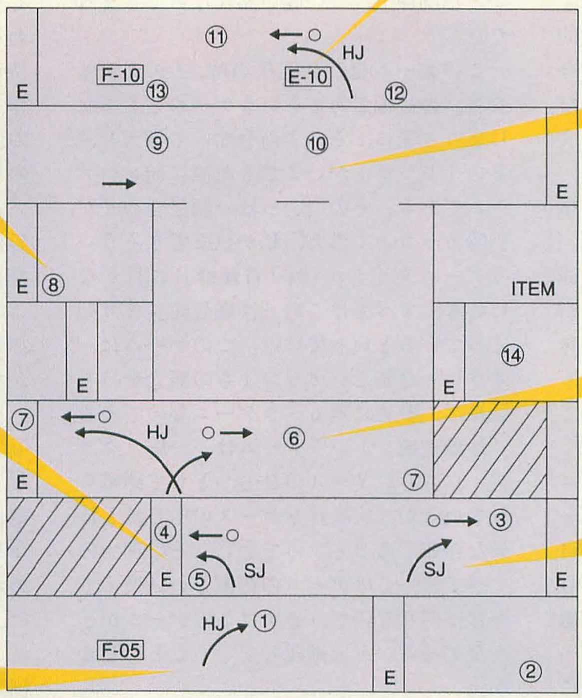
▶下手をすると敵につき落とされることがあるので、注意しよう



▶ベティーで空中の④を破壊したあと⑤の工作ロボットを料理してやろう



▶壁のライトの3つ目ぐらいでジャンプ斬り。タイミングを覚えよう

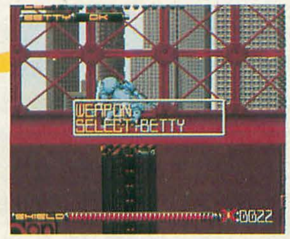




◀敵の弾をベティーで防ぎ、レバー上入力でベティーを敵に当てて破壊せよ



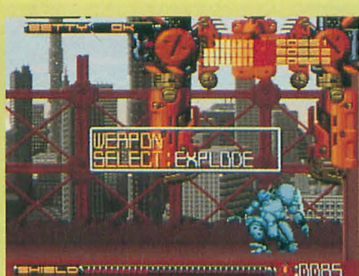
◀エレベーターから斜め上ジャンプで敵にベティーを当てて、すぐにダッシュ



◀ここでベティーを用意しよう。いよいよショウタイムの始まりだぜ！

STAGE2 BOSS

長いエレベーターを上るとボスが登場する。まあ，ガキじゃあるまいしエレベーターではじっとすること。意味がわからないのなら，試しに小ジャンプで両端ギリギリでバク転しなさい。すりぬけて落ちることになるから。ま，余談はここまで。まず右端にボスの右足がくるように位置を調整し，EXPLODEを準備。落ちてくる瞬間，あるいは弾が当たりそうなときに爆発させる。そして，わずかな無敵時間中にもボスの足を攻撃しよう。無敵時間が終わったら，すかさずベティーを使う。ちょうどこのタイミングでベティーが使用可能になっているはずだ(見事なバランス，これこそズームといえよう)。さっさと足を破壊したあとは，敵が位置調整を終えて落ちてくるときに右にいること。位置調整は写真のように，右側に空きがないようにすべし。あとは落ちてくれば自然とボスの上に乗れるので，ベティーを駆使して撃退しろ。本体が撃ってくる弾のスキをついて，ベティーを飛ばすのがポイントだ。あとは，破壊し損ねた砲台が残っているかもしれないことに注意すればいい。無駄なダメージを食らうことなく，効率よくボスを倒そう。



◀このタイミングでEXPLODEを使いボスの左足を破壊する



◀両足を破壊したあとは右端に寄り、ダメージ覚悟でボスの上に乗るのだ



◀振り落とされないようにしながら弱点にMAXベティーを叩き込め！

許された炎の龍

Yaegaki Nachi 八重垣 那智

どうしようもないんだけど，どこか憎めないヤツ
否定されればそれでおしまいのゲーム世界では結構特異な存在だ
そんな「沙羅曼蛇」の魅力を探ってみよう

●沙羅曼蛇



シャープ ☎03(3260)1161

言葉いじりの文化というものは，特にいつからあるようなものではなく，特定の国や言語に固有なレベルのものでもない。よくあるスラングや隠語などは，特定の目的に沿って活動している集団の中では，古今東西を問わず自然発生的に生まれるようだ。そういった言語いじりの中でも，本来の文法や語句を崩した違法な表現というのは，その違和感からくるインパクトからか，極めて印象深く認識されるように思われる。よく電車のガード下に「魔愚魔参上・世露死苦」などとスプレーで書いた力作などがある。こういった当て字に限らず，無意味な言葉で語尾を揃えたりといった簡単なものは，特に集団というものを意識しなくても意外に広範囲で使用されている。

これらは文明の根源要素でありコミュニケーションの基本であるところの「言葉」という社会常識の象徴的な存在を，故意に自由化(破壊)することで，制度や常識に縛られない自由な自己をアピールするという意識が働いてるように思われてならない。当て字や語呂合わせといった類は，文化的には比較的低級な領域に位置づけられているが，それは言葉という枠を破ろうとするアウトサイダー的なカッコよさと紙一重のものなのである。

目撃者のW

いきなり読者の9割ほどを，次の記事に吹き飛ばしてしまうような，ツカミにあるまじき話題で始まった今回のリバイバルレビュー。今回のお題は「沙羅曼蛇」ということで，最初にネーミングの話をしておきたかった関係上，つい硬い話になってしまった。

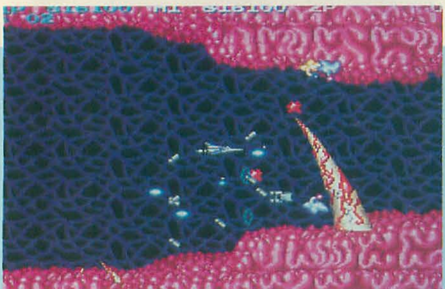
このゲームは，1986年当時，「グラディウス」の続編であるというゲームとしてなによりも先に，その名前を知って仰天し，そのインパクトがいまでも記憶に新しいゲームである。その「族」っぽい語呂合わせが，当時カッコよく思え，私がいまでもそういうゲームタイトル(例「首領蜂」)に目がないのも，すべてはこの「沙羅曼蛇」より始まっているといっていい。このゲームは，そういった新しいカッコよさの概念を持ち込んだ，歴史に残るべきゲームなのである。

「沙羅曼蛇」というゲームは，一応「グラディウス」シリーズのひとつとして認められているが，実際はシリーズの中で最も特異な存在であるといってよい。その理由のすべては，このゲームの特徴といい換えてもほぼ同じなので，まずはこのゲームがどんなゲームかを再確認していくことにしよう。

まず自機の操作系だが，移動のための8方向レバーにボタンが2つ。しかもショットボタンとミサイルボタンである。グラディウス最大の特徴であるパワーアップボタンがないことが，「沙羅曼蛇」の最大の特徴になっている。ちなみにパワーアップは，赤い敵を破壊したときに出現する特定のアイテムを回収すれば，ただちにそのパワーアップが装着されるシステムが採用されている。

パワーアップの自由度をゲームの魅力にしていた「グラディウス」とは，打って変わったシンプルさが，「沙羅曼蛇」を逆に特殊なものにしているのである。さらに操作系で特徴的なのは，「沙羅曼蛇」は専用の筐体でゲームセンターにデビューし，そのコンパネは2人分が横に並んでいたことであろう。つまり，これは2人同時プレイの採用を意味している。歴代の「グラディウス」シリーズで同時プレイが可能なのは，この「沙羅曼蛇」と「極上パロディウス」だけであることからも，その特異性は明らかだ。

もちろん操作が特殊なだけにゲーム自体もまた，ほかのゲームと比べてかなり特殊な展開をする。特にスクロール方向が横や縦にステージごとに変化する仕組みは，古今東西のゲームを探しても，ほかにひとつあるかないかというくらい独特のシステムだ。「沙羅曼蛇」は，横スクロールシューティングと縦スクロールシューティングの2種類のゲームを合体させて作られたものである。つまり，横スクロールシューティングの超代表作である「グラディウス」とは根本的にシステムが異なった代物なのである。ちなみに奇数面は横スクロールで，偶数面は縦スクロールと交互に設定されており，1周は全6面で構成されている。ただ，ステージごとに固有のイメージがあり，敵と戦い最後にボスを倒す手順は，シリーズの共通事項として継承されている。


地形にアイテムを埋めると最悪，慎重に狙え



いわゆる，あらえっさっさー状態。可愛い!?



敵の目くらまし攻撃。実はまったくの無害無臭



あらかじめ下がってくるところで待ち伏せよう



長いものには巻かれろ，でも１周目だけ

証言者のT

これほどまで違っているにもかかわらず，「沙羅曼蛇」は「グラディウス」シリーズの中のひとつである。それを納得させてしまうのは，これらのゲームの世界がつながっているからにほかならない。便宜的な設定上のつながりではない。美しいグラフィックと，耳に残るサウンド，迫りくる数々の敵に，息もつかせないゲーム展開，そのすべてがハッキリとは見えないものの，プレイヤーに同じ世界を感じさせるのである。だからこそ「沙羅曼蛇」は，「グラディウス」を継ぐことができたというのは，決して間違ったいい方ではないだろう。

しかし，「沙羅曼蛇」は当初は話題性もあったものの意外に早く人気が収束していった印象も大きい。少なくとも「グラディウス」よりその旬は短く，冬の寒さが気になる頃には，当時珍しかった専用筐体がゲームセンターで寂しそうにしているのを見ることも多くなっていった。ここでは，どうして鳴り物入りで登場したハズのこのゲームが，早々に求心力を失ってしまったのかを検証してみよう。

最大の悲劇は，このゲームが２人同時プレイというシステムを持っていたことにある。本来なら「沙羅曼蛇」の最大の特徴であるはずだが，それは必然的にその場復帰制というプレイヤーのミスに対するシステムが採用されるということにほかならない。いまでこそ珍しくないが，このシステムが緻密さを尊ぶ「グラディウス」のファンにとっては，極めてアバウトに感じられたに違いないと想像できる。

敵の猛攻の中で，ミスの直後になにごともなかったかのように初期状態で復帰する自機が，体勢を立て直すこともままならずに次々とミスを重ねていく。そんな状況は，たった１回のミスのペナルティとしてはかなり不適当といわざるをえない。この問題は，それよりも１年以上前に同じ２人同時プレイゲームである「ツインビー」においてすでに明らかだったはずだ。所持していたマルチプルのアイテムが画面に残るだけという中途半端な対策でごまかされていることは，よりその問題を深刻にしているにすぎないのである。

同様に緻密さに関する悲劇として，パワーアップ手順が，固定されているということを挙げることができる。これはパワーアップとして出現するアイテムは，すべてにおいてその出現位置と種類が固定されていることを指す。しかも前述したようにミスの許されないプレイが要求される中では，パワーアップが序盤で飽和してしまい，以降のアイテムは単なるボーナス点を与えるだけの存在でしかないのが実状だ。

またその種類と配置も偏っており，特に終盤でレーザーアイテムがまったく出ない

姉弟４コマ番外編 「サラマンダ」漢字で書くと「沙羅曼蛇」

弟：いまも昔も，２人でするゲームはありますよね。最近では格闘ものがその筆頭ですが，あれは，相手をあからさまに敵とみなしてプレイしなくちゃ駄目でしょ。そこが，平和好きな僕にはちょっと合わないのです。で，一般に協力型のゲームと呼ばれるソフトのひとつに，「サラマンダ」がありました。漢字だと「沙羅曼蛇」。
姉：ス・ピードアッツ。
弟：そうです。このソフトは，パワーアップ型シューティングの代名詞ともいえる「グラディウス」の系列でした。ですから，よく喋ります。
姉：ミッソウ！　モーサポ！
弟：姉ちゃん，静かにしてよ。仕事中なんだから。
姉：なにいってんの。このソフトで正しいLとRの発音を覚えるのよ。リッポーレイゾ。
弟：全然カタカナやんか。とにかくしばらくジッとしててよ。さ，さて。このソフトのもうひとつの特徴に，横・縦スクロール面が交互にあるということが挙げられます。これは毎回新鮮な気分が味わえて，個人的には気に入ってます。一粒で二度おいしいってやつですね。
姉：ハイブリッドを先取りしてたわよね。
弟：また，すぐそうやって覚えた単語を並べて。姉ちゃんの悪いクセだよ。子供が生まれたんだから，そのへんのところを直しといたほうがいいよ。それにしても，このソフトはお世辞にもよく作られたものとはいえませんでした（開発の方，ごめんなさい）。とにかく処理がとても重いのです。おかげで弾避けが実に楽になっています。きっと，世のX68000シューターは，あくびが出たんじゃないでしょうか。
姉：あんたはすぐやられてたじゃないの。
弟：う，うるさいなあ。僕はシューティングとかアクションは苦手なんだよ。X68000ユーザーがみんな鬼ゲーマーだと思ったら大間違いだよ。
姉：でもX68030 でプレイしたらまともな速度で嬉しかったのよねー。ただっ速。
弟：ちなみに「プリンス・オブ・ペルシャ」もスムーズになって感動しました。実は，時代を先取りのプログラムだったりして。
姉：やっぱり先取りのハイブリッドでしょ。
弟：はいはいそうですよ。さすが姉ちゃんだー。えらいえらい。
姉：……あんたは連射なしのパッドね。
弟：うぅ。姉ちゃんはすぐアイテムを横取りするし。
姉：２人同時プレイの基本は対戦よね，やっぱ。
弟：そうか，初めから，協力型だと思ってなかったんだな。いま初めて気づいた。うーん。

（岡村直也）





長いものには巻かれろ，その２。右上に注意



左に見えるは恐竜君。３周目の名物だ

ことには，悪意を感じてしまうほどの作為性が垣間見える。こうした「グラディウス」の特徴のひとつであった，パワーアップにおける作戦的な手法を許さないシステムが，プレイヤーに対してゲームの底の浅さを感じさせる原因になっているのは，疑いようもない事実だといっていいだろう。

批評家のC

また，「沙羅曼蛇」にはいくらか演出的要素も含めた，トリッキーな仕掛けや敵が多いことも，広い視野で見ると悲劇であると思われる。いわゆる３面のプロミネンスや６面の左右の壁についたミサイルといったものに代表されるところの「知識で対処するモノ」や「特定のパワーアップで倒す敵」のことである。

特に後者は，そういった敵が出現する前にミスしてしまった場合，即ゲームオーバーとなるに等しい状況を作り出してしまい，最初に挙げたこのゲーム最大の欠点を大幅に増幅する結果になっていることは間違いない。知識で対処するものについても，通常の敵配置や地形といったものならばともかく，６面のビッグコアに見られる脈絡のない安全地帯といったものを活用することが当然視されている。こういった構成からは，このゲームの上達が技術的な要素とはカケ離れた領域にあることを，プレイヤーに思い知らせる以外の結果を生まないことは確かである。

こうした些細なミスでゲームプレイ全体が台なしになるシステムと，プレイヤーの判断や工夫の入る余地のない構成が組み合わさってしまった「沙羅曼蛇」をプレイすることを考えてみると，単調なパターンプレイに容易に収束してしまうことが想像できる。正確に，安全にフルパワー装備を維持することだけに，ゲームが終始してしまうのである。何度も前と同じように，思い出しながら操作を繰り返し再現していくだけのプレイに，新たな発見を求めることは難しい。こうして「沙羅曼蛇」の魅力は失われていったのである。

これらの問題のいくつかは，翌年にリリースされた，３ボタン操作を採用して「グラディウス」方式のパワーアップを導入した，改良版である「ライフフォース」においても，いくらか軽減されていることからも，その深刻さを推し量ることができる。「沙羅曼蛇」が背負った，ヒット作である「グラディウス」を引き継ぎ，そして超えるための違いは，諸刃の剣として「沙羅曼蛇」自身にも大きな傷をつけてしまったということになるのだろう。

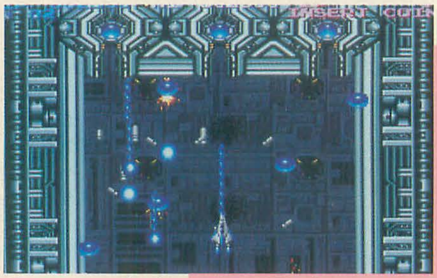
愛好家のL

こういった問題を抱えた「沙羅曼蛇」がX68000に移植されているわけだが，極力実物を意識して作られていることはいまさらいうまでもないことだろう。仕様的にはまったく一緒といってもよい形態での移植であり，当然ながら前述したオリジナルの欠点まで余すところなく移植されている。ゲームスタート時の自機数設定の部分にだけ，オリジナルとの明らかな違いがあるが，これについて問題になることは，まずないといっていい。

X68000移植版としての独自要素は，そういった姿勢からは見られないのであるが，ロード中に未使用音楽を流してみたり，ゲームを終了後電源を切るときに謎の声が出るといった，お遊び的な仕掛けが用意されていて，いい味を出している。

また，なんの必要があるのか知らないが，電源投入直後に行われる一連のセルフチェックの画面まで再現してあるのには，当初ビックリした記憶がある。しかし，それよりもなによりもこの移植版で驚かされるのは，このゲームが本物と比較して強烈に遅いという，移植という存在を失ってしまいそうな致命的な欠点にほかならない。

とにかく遅い。常に遅いのではなく，プレイヤーがフル装備で攻撃したりすると，その負荷に対応して遅くなっていくのである。スクロールも含めた，全体的な速度低下が恒常的に発生し，これに加えてハードウェア的な限界によるキャラクターの欠損といったものまで起こる。まるでそこには，同じ名前の別のゲームがあるように思えるくらい，違うモノが存在しているのである。特にその炎のアニメーション処理の美しさが最大の見所である３面で，無残に書き換えがガクガクと見えてしまうのだ。そのアニメーションしていることすら判別が困難になっているような状態を目の当たりにすると，オリジナルを知っている人ほど，この移植のギャップが大きいということがわかる。こうした欠点の積み重ねがX68000



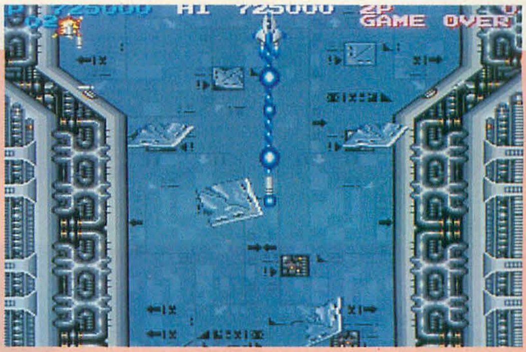
下のハッチを壊さないと青玉出現。痛いぞ！



地獄の突撃を抜けて，カモ相手にひと休み



懐かしやビックコア。後ろからPP7＋K(ウソ)



無敵の鉄板は剝がれる前に通り抜けるが吉

版の「沙羅曼蛇」を，駄作(クソゲー)だという認識に導いていることは，否定のできないところであろう。

しかし，こういった悪条件が重なっているにもかかわらず，この移植版「沙羅曼蛇」を遊んでいる人は多い。そして，皆一様にこの移植版が不十分なデキであることを指摘するにもかかわらず，この移植版を拒絶する人は少ないのである。かくいう私もそのひとりであり，普段から移植はプレイ感覚の再現を重視するべきだと口を酸っぱくして叫んでいるが，これを受け入れている。これはほかの人と同様，我ながら極めて不可解に思えてしようがない。ここにいたって問題になっているのは，オリジナルがよく知られたゲームにあるまじき問題を抱えた移植が，どうして許されるのかという点に絞られる。このX68000版「沙羅曼蛇」の最大の謎に迫ってみることにしよう。

分析屋のA

ここで「沙羅曼蛇」というゲーム本来の価値に立ち戻ってみたい。「沙羅曼蛇」というのはプレイの上達が知識的な部分に偏重し，正確なパターン再生プレイが要求されるゲームであることは，先に述べたとおりである。それはつまり，1秒でも長くプレイを，その正確なパターンを持続させることが，ゲームの快感に直結していることを意味している。

すると，「沙羅曼蛇」というゲームに対する満足は，自分が満足できる時間（たとえば1周クリア）に満足のいくプレイを維持できたかどうかにかかってくることになる。そこにはゲームプレイ中の一瞬の判断や駆け引きの積み重ねではなく，無事にそこまでプレイが持続したことそのものが，結果として与えられるという，パズルゲームにも似た，ちょっと特殊な満足があることになる。こうした状況においては，移植ソフトの技術的な問題に起因するような速度や表示の問題は，パターンの互換性が維持されてさえいれば，同様の行為に同様の満足が得られることで，無視することができるようになるのである。

これがX68000版の「沙羅曼蛇」における免罪符であり，これは名前こそ同じだが内容にまったく共通点のないファミコン版やMSX版ではありえない，唯一の存在価値である。つまり「沙羅曼蛇」をプレイするというスタイルを，このX68000版は再現することができるのである。たとえその速度が耐え難く遅かったとしても，画面のキャラクターが異なっていても，偽物よばわりされようとも，「沙羅曼蛇」というゲームに限っては，プレイするというスタイルが再現される限り，その移植が価値を失うことはないのだといってよいだろう。いい換えればX68000版の「沙羅曼蛇」は，本物に近い移植なのではなく，本物を想像してもらう移植なのである。

さらにそれ以降に発売された，プレイしている瞬間の感覚の再現を重視した，秀逸な移植ソフトの多くをも，この「沙羅曼蛇」は想像させていたのではないだろうか。そこには未来を想像させる，より完璧なものができていく過程の通過点としての役割もあったような気がしてならない。そういった意味では，この「沙羅曼蛇」はX68000における業務用ゲームの移植モノに，多大な影響を与えたソフトであると認識するべきであろう。オリジナルと同様に歴史に残るべきソフトなのである。

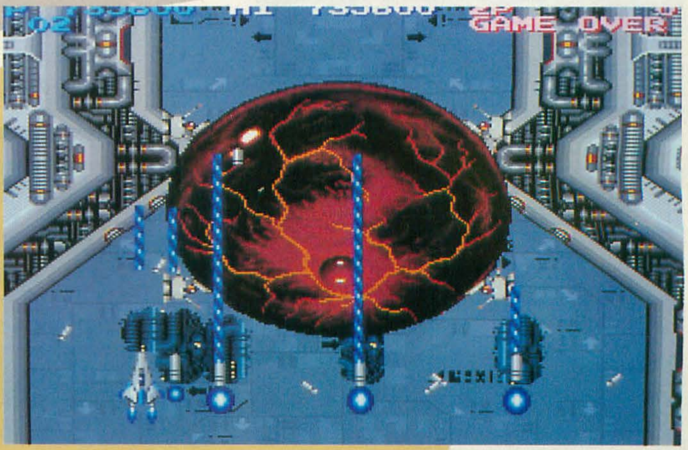
夢想家のV

いまとなってはなかなかお目にかかれなくなった「沙羅曼蛇」であるが，今回書いてきたようなことを踏まえてX68000版をプレイすることで，そのなんたるかを知ることはできるだろう。「沙羅曼蛇」というゲームを理解し，その移植の存在価値を確認することで，X68000のゲームがどういった発展をしてきたのかも，同時に知ることができるハズだ。

また，いまになってこうして振り返るということは，当時の未来への希望に満ちた純粋さを思い出すことでもある。そういった意味でも，「沙羅曼蛇」はX68000ユーザー必修のソフトなのである。



モアイよりも穴からの攻撃が危険！



ついに出た醤油卵型大ボス。弱点を攻撃しよう！

一転飛躍の年1988

Nakano Shuichi 中野 修一

そして朝日は昇る

先月に引き続きX68000黎明期のゲーム状況を見てみることにしよう。

1988年。低価格なX68000ACEが発売されてゲーム市場も勢いづいてきた頃だ。ゲームの発売本数もかなり多い。初年度は数本しかなかったので編集部にきたものは残らず手をつけていたのだが，2年目となるとさすがに全部やっているとは限らない。この頃になるとC言語も出ていていろいろ遊べるようになってきていたし，適切に解説できるかイマイチ不安だがとりあえずいってみよう。

2年目のラインアップをリストアップして見るといきなり超豪華に揃っているので驚く。ゲーム機が100万台，200万台って騒いでいるのとはわけが違う。2年目ではX68000なんてせいぜい5万台くらいだったはず（詳しい数字は忘れた）。早くも，「こんなに出して大丈夫？」状態に近づいている。しかし，圧倒的な勢いがあったので，大丈夫。1988年はイケイケ状態の年であった。

心配していたとおり，やったことないゲームの名前もちらほら見える。うーん，順番もかなりあやふやだな……。



「愛」に意味はないらしい



押し出しの強い安陀婆



ツインビー1面のボス



かわいいけど攻撃はきつい

1988のゲームたち

●ザコックピット

旅客機の着陸シミュレーションゲームである。3D処理で画面には夜景が広がり，コントロールタワーとの交信もそれっぽい感じ。しかし，他機種でも出てるし，これじゃあまりに地味だよなあ……とか，いいつつ意外にハマっている人が多かった不思議なゲームだ。しゃべるってのは結構よかったし，横風さえなければ結構無事に着陸できたりするものだ。プレイを分析してくれたり，点数がつくというのもいい。録音のNG集がでっかいPCMファイルで収録されていたりする。

●ツインビー

シャープ＆コナミのシューティングゲーム第2弾。同時に，

善「これってこのまま発売されたりしませんよね？」

編「まさか，そんなことはないでしょう」

という，善司ショックパターンの第1弾だったりする。その後，別のゲームでこれとまったく同じ会話が何度か繰り返されることになる……。

いまやシューティングゲーム以外のところでやたら有名になっているツインビーシリーズだが，これが元祖。もともとのアーケード版はかなりきついバランスの中にも，極めると神の領域に迫る美しい世界があったらしい。

それはともかく，X68000版ではファミコン版を移植するという暴挙に唖然とした人も多かったようだ。

●殺意の接吻

マンハッタンレクイエムの別シナリオ。基本的には前作と同じなのだが，ノリはさらによくなっている。シナリオディスクのみの発売というシステムも斬新だった。

●デスブリンガー

日本テレネットのファンタジー系RPG。

なにしろ私はこのゲームはやってなかったので，詳しいことはなにもわからない。名前からして面白くなさそうだと思うのは私の偏見かもしれないが，人に話を聞いても「とりあえず犬を殺してですね。キャンプするんですよ。するとですね……」と，



コミカルタッチの桃太郎伝説



大きなマップがうれしいA列車2

[1988]



最初はミジンコあたりから始まるたんば

バグ情報しか集まってこないようなゲームだった。きっとRPGだからストーリーとかも用意されていたんだろうなぁとは思っても，クリアした人の話は聞いたことがなかったのだ。

結局，詳しい内容は謎のままだったのだが，何年も経ってから「え，あれは面白かったじゃないですか！」という人が現れると手をつけなかったのがなんとなく悔やまれてくる。不思議なものである。

●源平討魔伝

X68000発表当時のナマイキ系愛読者ハガキでよく名前の見られたゲームで，デカキャラが動きまくる当時の最先端ゲームであった。表示はともかくキャラ定義数が足りず動くわけがないと思っていたら，あっさり動いてしまった。尼寺との違いを痛感するゲームだった。

●麻雀狂時代スペシャル

内容は脱衣麻雀なのだが，演出の勝利というべきか，「脱衣」でない見せ方がとても面白い。編集部内外でハマっていた人も多かった。かなりよくできたゲームではあるな，うん。

●桃太郎伝説

X68000で成功した数少ないRPGのひとつ。ファミコンからの移植だが，仕上がりはとてもよい。さすがハドソンというべきTITLE.SYSから始まり，適度にパワーアップされテカテカしたグラフィック，適度な難易度，完成されたシナリオを備えている。

発売後数年してから泉大介氏が「『だだぢぢの術』ってどう使うんですか？」と，なんの脈絡もない会話でポツリと尋ねてきたのが印象に残っている。使い方といわれてもねぇ。

●A列車で行こう2


アイテムを奪いあい昇天を目指す

いまでは世界的な箱庭作りゲームとなっているA列車シリーズ。A2までは最近の純シミュレーション風の味付けとは異なり，パズル要素が強い。なにより，大陸横断といった大きなスケールが魅力だ。個人的に私は最近のよりこっちのノリのほうが好きだ。

雲が流れるなどX68000版オリジナルのフィーチャーも盛り込まれている。電脳倶楽部ではユーザーの手による大名古屋マップなども発表されていた。一部では根強い人気があったようだ。

オープニング曲がボレロだったのだが，ラベルは1937年没だけど，フランス人だから……といったちょっとデンジャラスかもしれないゲームでもあった。

●たんば

完全な色物である。相原コージと高橋章子事務所の作った双六ゲームをコンピュータ化したもの。霊界をテーマに輪廻転生していくというものだが，あまり面白くないというのが欠点だった。

X68030用パッチ作成時のテストプレイで「あがり」のときにバスエラーが発生して牛島健雄氏の頭を悩ませていた。そういえば，もともとX68000でも2回に1回くらいバスエラーが出ていたんだよなあ。


対戦が楽しいドッジボール



このノリが命だ

●熱血高校ドッジボール部

とにかくノリがいい。ナッツシュートの攻撃テクと防御から攻撃へ転ずるキャッチテクとで技を磨き，くにお君とイガラシ君たち（編集部での通称）で世界制覇を目指す。

のちのボンバーマンと並んで，あちこちでどんな大作ゲームにも劣らないくらいの圧倒的な支持を得ているゲームだ。

ロンドンの石畳，アイスランドの氷の大地，天安門広場，アフリカのジャングルといった国際色豊かなステージ構成で，ぜひ輸出禁止品目に指定すべき逸品。編集部ではいちばん端で横になっているペンギンに人気が集中していた。

やはりこういうのは対戦が熱い。対戦時のキャラクターやステージセレクトもほし



ニトロで500km/hに加速するフルスロットル



コースはさまざまに分岐している



えーと誰のことだろう……



マルチウィンドウは好みが分かれるか

かったところだが……。

●フルスロットル

なかば伝説と化したドライブゲーム。

残念ながら私はアーケード版を知らない。最初はなんてタイム制限がきついんだろうと，ぜいぜいいって1面をクリアしていたのだが，実はずっとローギアで走っていたという……。それはともかく，走るほどになくなるスピード感，敵車のアルゴリズムは単純で悪質と，まったく爽快感のないゲームに仕上がっている。X68000XVIで遊ぶと別次元のゲームになってそれなりに面白いのだが。

実は発売を一度延期して作り直されており，私はどちらにも触ったことはあるのだが，前後のバージョンでどっちがよかったかといわれると，……甲乙つけがたしというしかない。

これ以降，どうも車ゲーには恵まれないようだ。

●ソフトでハードな物語

パソコンソフト業界をテーマにした新感覚のアドベンチャーゲーム。簡単操作で，肩のこらないシナリオで（?），業界のことがよくわかるというオマケつきだった。内容はまったくのフィクションだが，妙にリアリティを感じるのは気のせいではないかもしれない。クラシックアレンジによる音楽が飛び抜けていい。

●琥珀色の遺言

リバーヒルの並々ならぬ気合が感じられるアドベンチャーゲーム。とにかくため息が出るほどオープニングが美しい。半透明2重スクロールするタイトルには度肝を抜かれた。オープニングを見るためだけに立ち上げたものだ。残念ながら，中身はやってない。ごめん。

●信長の野望　全・国・版

光栄のX68000参入第1弾。結構力が入っている。なにしろ，しゃべる，効果音は入る，なによりフルマウスオペレーションである。「あの信長」もX68000ではこうなるのか！　という感慨に胸を打たれていた人は多い。

しかし，これ以降，光栄の移植作品は時期を追うごとに気合が抜けていく様がよくわかる。他機種との共通化で開発効率はよくなっているのだろうけど。

●ドラゴンスピリット

正直いって，アーケード版が出たときは色物ゲームかと思っていた。自機がドラゴンで，羽根の先まである当たり判定，同時に押すと弾の出ない地上/空中別攻撃。動く地形……。

しかし，実際にやってみると音楽や効果音の素晴らしさ，キャラクター，特にボスキャラの多彩さ，古代生物をモチーフにした世界観に驚く。当時，X68000を持ってなかった金子俊一君が速攻でソフトを買い，日々パッケージを愛でていたというのもわからなくはない（そーか?）。

最終面が手ごわく，当初は断念していたのだが，買ってきたゲームでクリアしてないものが残っているのもしゃくなので，約3年後に突然ドラスピに目覚める。

無事アリーシャ姫を救出。

「アムルーぅぅぅ」

いやぁ，恥ずかしくていいなあ。

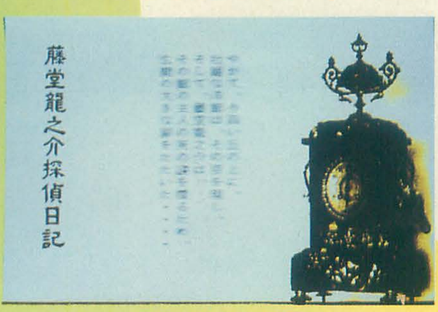
電波新聞社も縦画面を用意するなど力の入った移植を見せてくれた。OLD ROMとNEW ROMの2バージョンにも対応している（世の中にはOLDでないとダメという人もいる）。

このゲームとは離れるが，この頃から，同時期に同種のゲームが重なるという現象が起き始めていた。この時期にはシューティング3つがほぼ同時に発売された。堂々の沙羅曼蛇に，名作ドラスピ，力作のサンダーフォース2である。そういえばドラスピはほかの2つに比べればあまりしゃべらないなあ。まあ，「首みっつぅ」とかいわれても困るか。

●沙羅曼蛇

グラディウスの傍系にあたるシューティングゲーム。ほかのページで八重垣氏が書いていると思うが，なんやかんやいって，「綺麗だからいいや」で許されているゲームだ。ゲームセンターでは聞こえなかったBGMもなんとか聞こえるし……。

プログラム的にはフォースフィールドが4個までつけられるはずなのだが，2個つ



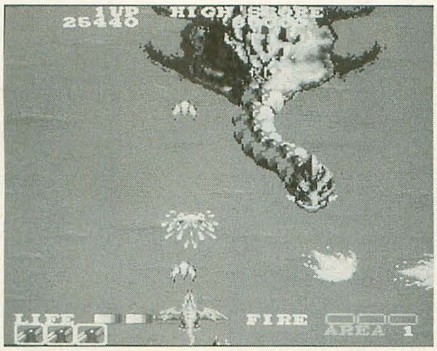
オープニングが圧巻の琥珀色の遺言



ゲーム部分は他機種とあまり差はない


いきなりマウス対応の信長の野望

[1988]


多関節くねくねの１面のボス

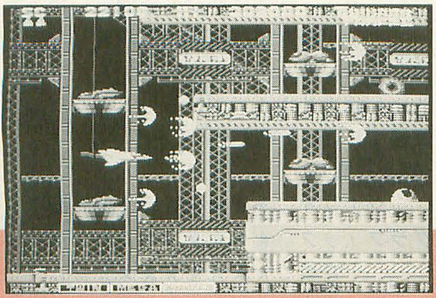
けるのも至難の技といわれている。１周で２個しか出てこないのだから，事実上３個は不可能だ。低難易度で家庭用として発売されたので２個ついたとこが見れるか？　と期待したのだがいまだ見たことはない。

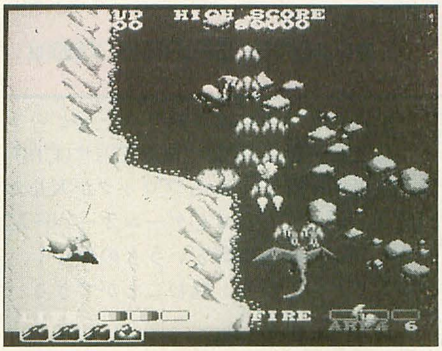
年末の忙しい時期にシューティングが重なったためかOh!Xでの扱いは３本まとめて４ページのレビューという冷たいものであった。祝一平氏に「沙羅曼蛇どうぞ」ってのをやってほしかったところだが。

ちなみに，たしか，このゲームが善司ショック第２弾だったと思う。

●サンダーフォース２

X1ユーザーにとってサンダーフォースというのは特別な意味を持つゲームだ。その第２弾がX68000でパワーアップしてやってきた。さすがにテクノソフト，いきなりの参入にしては完成度が非常に高い。８方向スクロールと横スクロールを組み合わせたゲーム構成で半透明２重スクロールなど，X68000の特殊機能を素直に使っていることがかえって新鮮だった。アーケードゲームをこなす力を持っていることはすでにわかっていたが，それに独自の機能を加えれば強力になるのは間違いない。


横スクロール面はかなりきつい


雪原のアザラシがいやらしい

自機の移動速度が速く，まだシューティングに慣れていなかった私にはとてもつらいゲームだった。また，乗り物酔いしやすい私には別の意味でつらいゲームでもあった。

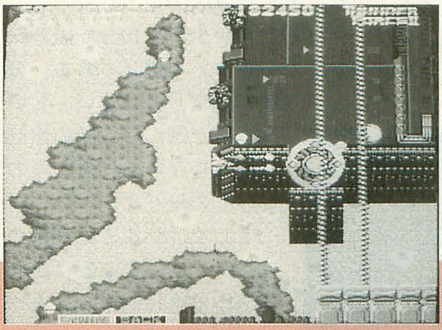
とはいえ，この年の年末に発売されたシューティング３本のなかではビッグタイトルに囲まれ苦戦を強いられたようだ。X68000オリジナルでユーザーの期待は高いとはいえ，いちばん割のあわなかったゲームかもしれない。

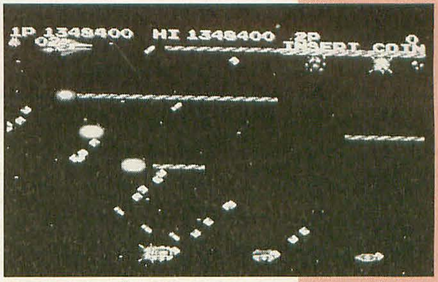
●テトリス

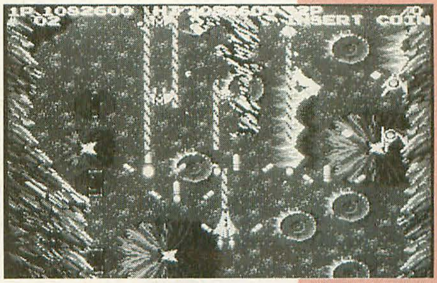
いわずと知れたテトリスである。一応説明しておくとアーケード版ルールではなく，オリジナル版の流れを汲む正統派だ。

定番ものではあるが，いまひとつキャッチーなところがなく，ちょっと弱かったかもしれない。

Macintosh版に比べてもいまひとつのめり込めない，なにか致命的な欠点があったような気がするのだが，それがなんだったか思い出せない。とりあえずブロックは綺麗だが……。


多重スクロールで酔いそうになる……


ほっておくと弾の嵐……

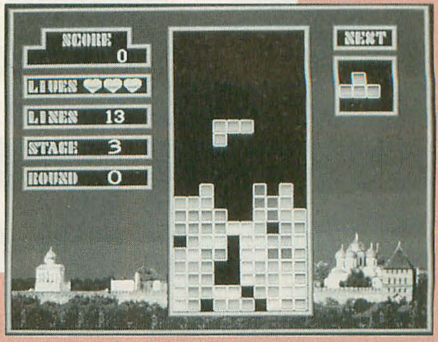

レーザーで岩を掘り進む

まとめ

この時期になるとX68000購入者の大半はゲームを目的のひとつにしていたと思われる。世間ではゲームばかりしていると思われるのも無理はないが，各種ツール類も好調だったし，ユーザーの怪しいプログラムもどんどん増えてきていた時期だ。まさに昇り調子といえるだろう。

ゲームの発売本数も徐々に増えてきた。今回は手早くまとめてみたのだが，さすがに全作品を網羅することさえもできなかった。来年分（1989年）はもっと増えてくるだろうから，私ひとりの手には負えないかな？

多少心配しつつ，来月へ続く。



BGMはコロブチカで決まりだ

THE USER'S WORKS in TAKERU

●ほいっぷるX68k/Devil's letter of Invitation●

ほいっぷるX68k ●PASTELSPIRITS/X68000



落ちものパズルならぬ押し上げパズルゲームがこの「ほいっぷるX68k」だ。とはいっても落ちもの系パズルゲームと同じく，ルールは簡単。2個ひと組みのブロックを左右に移動し，適当な位置でスペースキーを押すと，そのブロックが1ライン分上がっていく。そして，上がった先の横1ラインに同じ種類のブロックが，3個以上含まれている(隣接していなくてもいい）とそのブロックが消滅する。空いたスペースには，積み上がっていたブロックが落ちてくるという寸法だ。もちろん，落ちたときに横1ラインに同じブロックが3つあれば消滅する……というような連鎖が起きる。高い位置で消せば消すほど高得点を得られるが，ブロックが天井まで積み上がってしまうとゲームオーバーだ。

ちなみに，制限時間というものはなく，心ゆくまでのんびりと悩むことができるパズルゲームである。

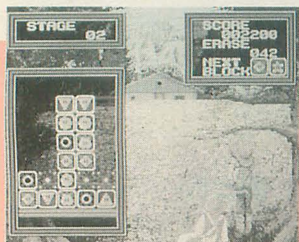
そして，ゲームモードは，

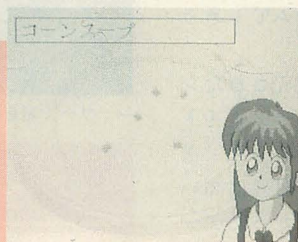
・NORMAL MODE：一定個数のブロックを消すことでレベルクリア。ただし，レベルが上がってもノルマが上がるだけで，ゲームの難しさ自体はそれほど変わらない。最高レベルは8。

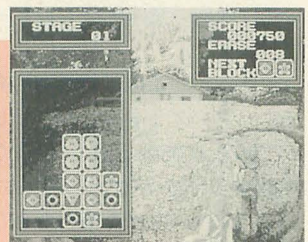
・RANKING MODE：ひたすらハイスコアを目指して修練を積める。

以上2通りが用意されている。

とにかくほのぼの～とした雰囲気がかわいい作品だ。





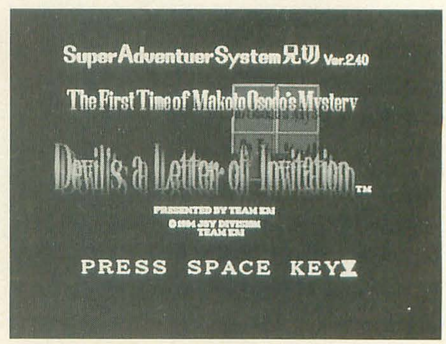


価格：1,000円(税込)

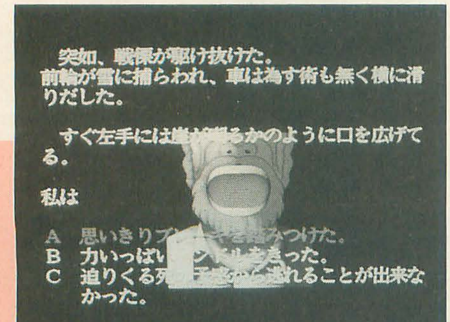
〈総合評価〉

ゲーム性：★★★★★★☆☆☆☆
料理：★★★★★☆☆☆☆☆
音楽：★★★★☆☆☆☆☆☆
お買い得度：★★★★★★☆☆☆☆

Devil's letter of Invitation ●Team K&I/X68000

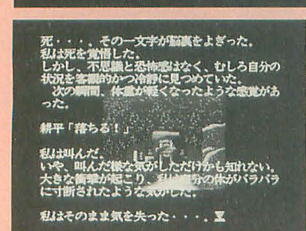


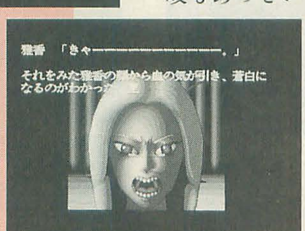
主人公はスランプに陥った推理小説家虹川耕平。ある日彼は，学生時代の友人から「どうにも不可解な体験をした。ぜひともこのことを作品化してほしい」という内容の手紙を受け取った。気晴らしに誘いに乗ることにしたのだが，道中吹雪に見舞われ……。



このような感じで始まる「Devil's letter of Invitation」は，1本のストーリーを楽しむノベルウェアタイプのゲームだ。ジャンルはサスペンスもので，ゲーム中にはいくつか選択肢があり，プレイヤーの選んだ行動によって多少テキストが変化するようなシステムになっている。

ただ，全体的な量が少なく10分で読み終えられるような短編という感じである。分岐もあっという間に収束してしまい，どの選択肢を選んでも話の本筋は変わらないのはちょっとだけもの足りない。





そして，小説という割にはグラフィックが表に出すぎている感じがする。あくまでも文字情報をメインとして，グラフィックはプレイヤーの想像力をかきたてるイメージ的なものをバックに敷くくらいが，ちょうどいいと思われるのだが。あと，音楽はかなり雰囲気に合っていて，話を盛り上げる演出効果としては十分な出来だろう。あとはもう少し，ストーリーを読ませる工夫がほしいところだ。

価格：1,700円(税込)

〈総合評価〉

シナリオ：★★★★☆☆☆☆☆☆
グラフィック：★★★★☆☆☆☆☆☆
音楽：★★★★★★★☆☆☆
お買い得度：★★★★★☆☆☆☆☆

［特集］

Now Printing........

任意ドット数での解像度変換や誤差拡散による減色法といった印刷の基本的なアルゴリズムはすでに確立されている。
不完全ながら，色変換についてのアプローチも行われている。
とはいっても実はこれらはすでに4年前の技術である。
プリンタの進化にともなった解像度の向上は圧倒的な印字品質をもたらした。誤差拡散法は低解像度のときに有効な減色アルゴリズムであったが，拡大して何ドットも同じ色が続くような場合には必ずしも最適というわけではない。印字品質はともかく，処理速度で不利になってくる。X68000での印字では解像度はある程度抑えてから多階調での誤差拡散を行い，あとは濃度パターン法で埋めていくというのが効率がよい方法であろう。さらに高次の印字ということになると，元画像の情報量を上げるしかなくなる。X68000の512×512ドットではもはや役不足になりつつあるのだ。大きな元絵を用意するというのがいちばんなのはいうまでもない。しかし，拡大処理で発生する画像の荒れを少なくし，画像からできるだけ情報量を引き出すということも考えるべきだろう。

最近ようやく最終兵器的なフラクタル圧縮の解説書が現れてきた。
――フラクタル解析。理論上，この方式を用いれば任意にディテールを補間しつつ拡大することが可能になる。フラクタル解析は圧縮目的のみならず，画像拡大ルーチンの最高峰となるべきものであろう。問題はキーとなるアルゴリズムが各種特許で保護されているということだが……（公開はされている）。
現時点で考えられるものからさらに進んだシステム……。
次世代のハードコピープログラムというものは，おそらく，そのようなものになるのであろう。

CONTENTS

[特集]

シャーペンの活用

根性の文字出力

Nakano Shuichi 中野 修一

ここではシャーペンによる文字出力について考えよう
恒例だが，タイトル周りを除いてこれらのページはシャーペンで出力されている
いろいろな問題点は根性で切り抜ける。DTPへの道は遠いだろうか？

印刷といっても，一般の人はグラフィックを扱うより文字を打つことのほうが遥かに多いはずだ。

シャーペンの登場により，日本語入力環境は格段に整備された感があるものの，出力環境はあまり恵まれていなかったといえるだろう。印刷表現が非常に限られていたので（ただしTEXは除く），使いこなしという面では突き詰める余地がなかった。

ワープロやシャーペンでは，普通の人が使う場合ならいくつか打ち出してみて自分の目的にあった書式を見つけると，あとはそれをちょっとアレンジしていくくらいのものだろう。テクニックといっても，せいぜい罫線をどう使うかという程度のことだ。

そのあたりもシャーペンのバージョンアップで少しずつ変わろうとしている。特にシャーペンワープロパックで印刷環境はかなり改善された。近日発売のシャーペンワープロパックのver. 2.0が出れば（すでに出ているかな？），かなりの問題が解決されるだろう。

問題なのは，現在手もとにワープロパックver. 2.0がないことだ。サンプル版もこの記事には間にあわないだろうというのは非常に不幸なことであった。ここでは，とりあえずver. 1.0をもとに，シャーペンでの文字出力や図版の扱いのテクニックについて解説してみたい。

シャーペンのテクニック

正しいインライン変換，多彩なエディット機能，さまざまなカスタマイズと拡張を可能にしたシャーペンは，プレーンテキストを作成する環境としては申し分のないところまできているといえるだろう。

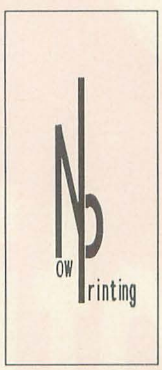
しかし，完成した文書を作成するということでは無敵のシャーペンも，当初はなぜか出力周りが非常に貧弱であった。ワープロとの差別化ということであろうが，その後も機能は整備されるものの，たとえば，段組み印刷がサポートされても，それは用紙の余白を無駄にしないためのものといった感じが強く，段組みのレイアウト文書を作るための機能としては弱いものがあった。完成した文書を作り上げることは苦手としているといえよう。やろうと思えばかなり凝ったものが作れることは確かだが，イメージどおりのものを作るのは難しいものがある。

しかし，いくつかの制限があっても，シャーペン自体の機能は非常に高い。その裏をかいていろいろなことをやるのも一興である。「シャーペンでDTPを行う」というのを目標とし，それを阻む要因となっているものを挙げ，対策を練ってみよう。

イメージの張り付け

Easydrawで作成した図はコピー時に余計な空白を入れてくれるので，シャーペンでの扱いがかなりやっかいになっている。

次のような図の場合，最小範囲でコピーしても，



とここで四角で囲っただけの余白を持っている。

ざっと見ると，これくらいならそれほど不適当な大きさではないようにも思えるかもしれないが，往々にしてこれが非常に厄介なものになってくるのだ。

シャーペンでは文字の左上隅を基準に位置を指定するので，特に文書の右端や上端に図をあわせたいという場合には苦労することになる。こういうので困ったことのある人はいないだろうか？

ある程度大きな図を作るときには，こういう処理は不可能なもんだと思っている人もいるかもしれない。

まず，図の左寄せには「文字間ピッチ調整」を使用する。この設定にはマイナスの値も許されているのだ。ただし，その行に図のイメージしかないと文字間というものも存在しないので，無理やり文字間を作ってしまおう。行の先頭に空白を1個入れ，次にイメージを張り込み，行ごと文字間ピッチを調整すれば目的は達成される。

ちょっとやっかいだったのは図の上あわせである。縦方向の文字送りを決定するのは「強制改行幅の設定」だが，強制改行幅には0以下の値が許されていないのだ。

これを解決するにはワープロパックが必要である。手順は以下のとおりになる。

1) まずページの先頭にあわせて文字オブジェクトを新規作成する
2) その文字オブジェクトを「縦書き」に設定する
3) 空白を1個置きイメージを張り付ける
4) その行の文字間ピッチをマイナス値で設定し調整する

ただし，最初に大きめなオブジェクトを設定しておかないと微妙な位置調整ができないことがあるので注意。あとはちょっとした応用でなんとかなるだろう。

あと，左端と上端にかかった画像はクリッピングされるので注意が必要。右端や下端の場合は画像がそのままはみ出して印刷されたのとは違う状況となる。しかし，タイトル周りなど，この部分にはみ出して印刷したい事例は結構多いのも事実だ。

そういった場合の手段として，上下反転が考えられる。

すなわち，そういったものが必要なページ内だけ反転処理をする。まず，「改行挿入」で各行を分割し，ページ内の上下を入れ換える。キーボードマクロなどを用いればさほど難しくないだろう。さらに，各行の文字列を左右入れ換える。これもキーボードマクロを活用すれば比較的効率よく処理できる。

。だ実事ものい多構結は例事いたし刷印てし出みはに分部のこ，どなり周ルトイタ，しかし。るなと況状う違はとのたれさ刷印てし出みはままのそが像画は合場の端下や端右。要必が意注でのるれさグンピッリクは像画たっかかに端上と端左，とあ

のように文章を反転させておき，

あと，左端と上端にかかった画像はクリッピングされるので注意が必要。右端や下端の場合は画像がそのままはみ出して印刷されたのとは違う状況となる。しかし，タイトル周りなど，この部分にはみ出して印刷したい事例は結構多いのも事実だ。

といったように，フォントを回転させて対応するという手だ。ただし，これだと部分的に行揃えを調整しないとおかしくなることもあるのだが，いたしかたない。

はみ出す方向が多方向な場合に使えないという問題はあるが，これでひとつの問題がクリアできる。

ただ，やっぱりかなりしんどいので，この原稿の出力でもこの方法は使っていない。1発出力にこだわらなければ，後述の2度打ちで印字位置を試行錯誤するほうがまだ実用的であろう。

しかし，ワープロパックver. 2があったら，こんなのは不要な技術なんだろうなあ……。

フォントの話

文字だけならばプリンタの内蔵フォントを使ったコード印字がもっとも美しい出力を得ることができる。

よく考えるとこれはおかしくないだろうか。プリンタの解像度自体はイメージ印字でも変わりはない。コード印字IFMの搭載により，さまざまな書体や大きさの文字を駆使して文書作成ができるようになったはずだが，印字品質はコード印字のほうが高い。要するに内蔵フォントの品質とIFMフォントの品質の問題である。

IFMは初期バージョンの速度が尋常じゃないほど遅かったということで，なかなかフォントの問題点が浮かび上がってこなかったのだが，フォントエディタを使えば，データの品質があまり高くないということがわかる。

図1を見てみよう。これがIFMフォントAmadeusのAのポイントデータである。あちこちにガタついたところがあるのがわかる。これではある程度以上に拡大することは無意味だ（スムージングやベジエ化などはフォントの品質を下げるだけである）。SX明朝なども形状には問題なくても，まったく無駄と思えるポイントが多くデータを重くし，下手をするとフォントの品質を下げかねない結果になっている。アウトラインデータだから直ちに高品質なフォントと考えるのは非常に愚かしいことがわかるだろう。

先ほどのAmadeusを修正したものが図2である。このようにデータを修正していけば，アウトラインフォントが本来持つべき品質のフォントを手に入れることができるはずだ。あとはもう根性だけである。

さて，ここで活用されるのが，フォントエディタ書家万流だ。印刷まで考えた場合，シャーペンユーザーにとって，Easydrawと書家万流は必携のツールである。

書家万流は5月号でちゃんと「使いものになるツール」として書いたはずだが，メーカーの人には多少誤解されたようだ。使えるツールでも操作性の問題やバグは放置すべきではない。それにこういうのはちゃんと使えば誰にだってチェックできるものなのだが。

なお，一部で行われていたような「インレタを取り込んでフォントを作る」ってのは，危ないのであくまでも個人的な用途に留めるべきだろう。

ということで本題に入ろう。

美しいフォントさえあれば，アウトラインフォントシステムは十分に使いでのあるものである。で，どうすれば美しいフォントが手に入るかというと，自分で修正していくしかない。

問題は半角文字ばかりではない。幸い，書家万流によって，標準アウトラインフォントが供給されることになった。これを基準にフォントをエディットしていけば効率よく好みのフォントを揃えていくことができるだろう。

ちなみに，ここまで使用されてきているフォントは，私が平仮名部分を書家万流でエディットしたものである。

比較のため，元になったSX明朝を次の2段落に使ってみよう（よく見ること）。

DTPではフォントは非常に重要な意味を持つ。たとえDTPツールがあったとしても，経験的にいって，フォントの品質がよいか悪いかでなにか作る気になるかどうかというのは大きく左右される。特に基本明朝体とゴシック体はできるだけ癖のないものを

図1　AmadeusのA

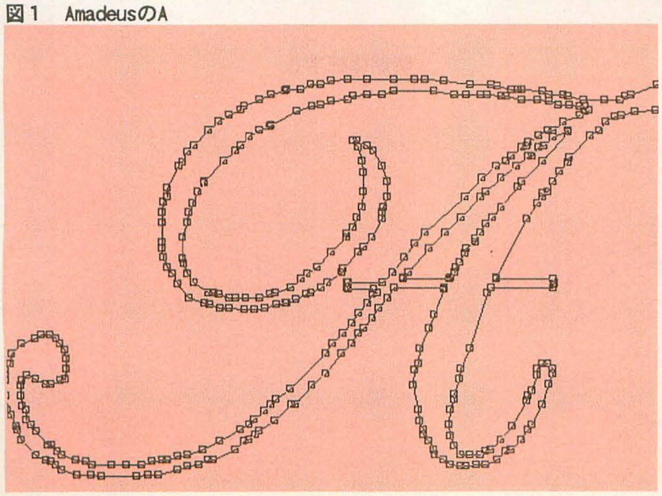
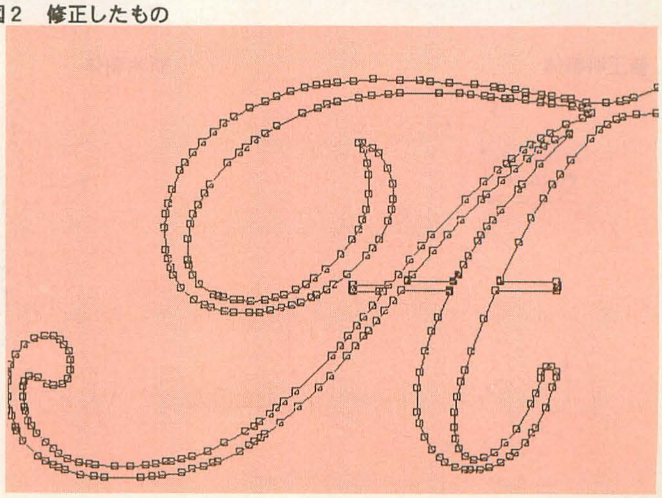


図2　修正したもの



使いたいと思う人が多いようだ。ちゃんとしたDTPというものを確立するには，多くの人が馴染んでいる石井明朝などの写研フォントが使える環境というのが最低条件といえるかもしれない（UNIXベースならあるらしい）。ほかのフォントで代用できないかという話もあるが，PostScriptやTrueTypeで発売されているフォントの多くはポップすぎて使えないものが多い。常用できるものという意味では，ゴシック系のフォントはほぼ全滅である。写研流でいうとほとんどが「ゴナ」系のものになっているのだ。

基本になるSX明朝を見てみよう。まず，この手の文字としては大きめに作ってある。ほかのフォントとの組み合わせなどを考えてもこれはいいことだ。さらに，限りあるスペースを有効に使うためか，隅々にまで文字が広がり全体のフォルムも四角くなって，ノリは新聞系ないし丸文字系に近くなっている。なにか気合が入りすぎていて見てるとなにか疲れてしまう。常用するにはちょっとつらいかもしれない。もっと自然体の文字がほしくなる（以上，SX明朝）。

そこで，最終的にはフォントを自分で作成しようということになってくるだろう。しかし，全体を作り直すのはさすがに手間がかかりすぎる。が，フォントのなかでも多用される平仮名部分を変更すれば全体の印象はそれなりに変えることができる。

実際に平仮名部分をエディットして癖を少なくし，英数字のプロポーションを変えたものがこのフォントだ（修正明朝）。一応，常用フォントを目指してエディットしたものだが，フォントデザインは完全に私の好みによっている。自分で作れば少々おかしくても納得がいくというものだろう。

SX明朝の癖から抜け出ていない部分もあるが，平仮名部分の多少の変更で文章の印象は結構変わってくることがわかるだろうか。ちなみにカタカナや漢字部分はSX明朝と同じである。全体にもう少し細くしたほうが読みやすいのだが，ボールド化のアルゴリズムはイマイチ信用できないのでここでは太めのままだ。

比較のために平仮名を並べてみよう。上がSX明朝，下が修正明朝である。

いろはにほへとちりぬるをわかよたれそ
いろはにほへとちりぬるをわかよたれそ
つねならむうゐのおくやまけふこえて
つねならむうゐのおくやまけふこえて
あさきゆめみしゑひもせすん
あさきゆめみしゑひもせすん

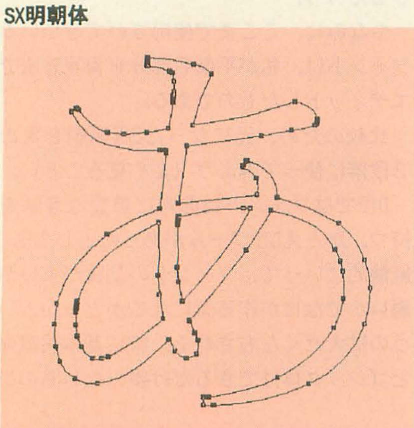
これが漢字交じり文だとどうなるかは，この原稿全体を見ればわかるだろう。

ほかの明朝体も集めて見比べてみよう。解説はそれぞれのフォントで行っている。

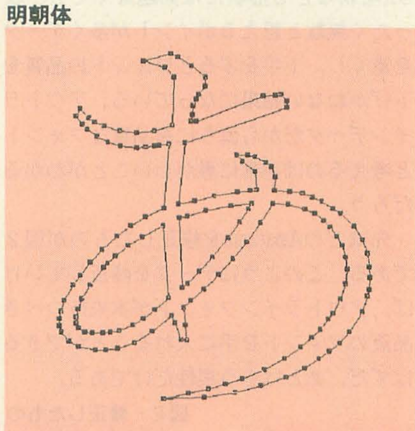
まず，ID＝512，514の明朝体。旧書体倶楽部，Z′sSTAFF ver.2.0に付属していたものである。Z′sSTAFFについてきただけあって，X68000での普及率はかなり高めだと思われる。一般には秀英明朝と呼ばれている書体だ。特徴を見てみよう。基本的なフォルムは見慣れた感じのものだが，平仮名で文字のつなぎが長く持続しており，1文字1文字を見ると実はそれなりに癖が強い。文章で見ると意外と読みやすく，違和感が少ないのは全体に細めで，それぞれの文字のバランスがよいからだろう。ただし，漢字は文字面を大きくしようとしすぎているきらいもある。直線補間のデータ形式でもフォント品質にはほとんど影響がない。細めのフォントは本文に使いやすい。

続いて，ID＝516の新明朝体。Z′sSTAFF ver.3.0に付属していたフォントだ。ver.2.0に付属していた明朝体と比較すると，1文字ずつに非常に癖が少なく，平仮名の文字のつなぎなどは逆にストイックなくらいだ。全体の文字面はやや小さめか。漢字はやや太めで，確かにグラフィックツールで使用するにはこういったデザインの書体のほうが向いていると思われる。しかし，

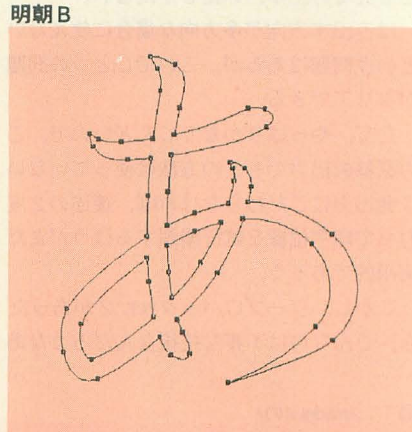
SX明朝体



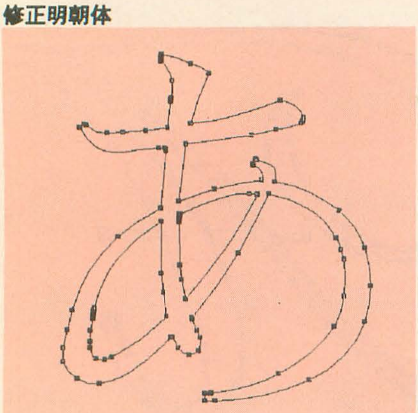
明朝体



明朝B



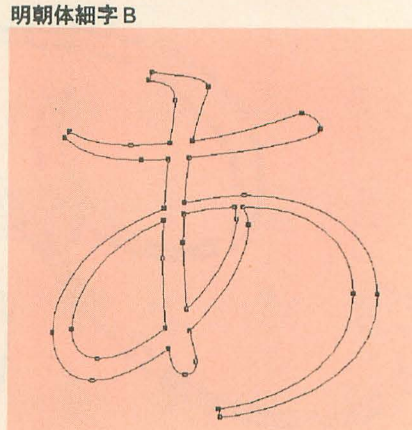
修正明朝体



新明朝体



明朝体細字B



フォント自体のデザインはともかく，データを作成状況に問題がある。曲線がゴツゴツしてアウトラインデータの品質は低く，拡大文字には適していない。それでも常用文字としての用を足しているのは見てのとおりである。拡大以外では万能に使える書体といえるだろう。

ID＝601の明朝B，JGフォントの基本明朝である。どちらかというと教科書体よりで，美しいと感じるかどうかは個人差があるかもしれない。さらっと流していながら，線のメリハリが強く，よくいえば非常に洗練された匂いのするフォント，悪くいえばデザインされすぎたフォントである。SX-WINDOWとは離れて，ある種のチラシやポスターで使用するには最適だろうか。文字面が小さいので文章用として使うには必ずしも適さないかもしれない。また，半角文字はイマイチフィットしにくいように思われる。個人的には好きなフォントだ。

最後に明朝体細字B。JGフォントの明朝体セットからの参加だ。フォントの拡大図を見ればわかるとおり，データを作成したフォントグラファーの技量は圧倒的に高い。ベジエ曲線をちゃんと使いこなしていることがわかる。このポイント数の少なさでちゃんと用件は満たしている。実に見事だ。フォントデザイン自体は丸文字系の色が出ており，これも好みが分かれるところだろう。文字面は大きめ，フォントは細めなのでほかのフォントとの相性も悪くない。常用で使えるフォントといっていいだろう。ただし，1文字ずつでまとまりすぎていると，文章中で使用したときに少し読みづらい。文章で使うフォントは多少崩し気味のほうが流れが見えてよいような気がする。

＊　＊　＊

シャーペンでもこういったフォントを使用できるようになって，実際にこのような文書が作成可能となっている。明朝だけでもこれだけのバラエティがあり（使い分けることは少ないだろうが），人の好みというものも多彩である。ぜひ，自分好みのフォントを作成してみてほしい。

大きな図を入れる

たとえば，この文書のように3段組みの書式で作成している書類に，段をぶち抜いた図を入れたいという場合を考えよう。前ページのAmadeusのフォントパターンのようなものを置きたい場合だ。

すでに文字間ピッチを使って図を移動させる方法については解説してあるが，図の大きさが字詰めよりも大きくなると，「文字の次にパターンを置いて文字間を作る」というのができないかに見える。パターンを置いた途端に改行されてしまうからだ。

こういう場合はかまわず前の行ごと範囲指定してから文字間ピッチを設定すれば大丈夫だ。なお，その図が入るところはちゃんと空けておくこと。プレビュアがないのでテスト印刷は念入りに行うべきだ。

しかし，極度に横長の図の場合は扱いがかなりうっとうしくなるため，可能ならば分割して張り込んだほうがよいだろう。ドローデータの場合はともかく，ビットマップデータの場合はこのへんはどうにでもなる。図1，2は実は1枚絵だ。

2度打ち

シャーペン上では，やろうとさえ思えばいろいろなことができるはず，とはいっても，さまざまな制限があって面倒なことも多い。凝った処理をしようとしても，文書ファイルが込み入ってくると，必然的に編集作業も重くなり，作業効率もどんどん落ちてくるだろう。

そういったときにおすすめなのが，パーツごとに分けておいて，一度出力した用紙を2度打ち，3度打ちすることである。

文書は文書，図は図，タイトル部分も別に……とやっていけばそれぞれの工程は単純化される。

図などを組み合わせたときのイメージがつかみにくくなる可能性もあるが，それはテスト印刷でなんなく解決できる。

ちょっとした図程度ならこんな大げさなことは不要だが，図版や地紋などを多用する場合はきわめて有効な手法となる。最近のプリンタの精度はそれなりに信用していいので，多少のコツをつかむだけで複雑な処理もできるようになる。

この手法からさらにレーザープリンタとインクジェットプリンタを組み合わせるというアプローチも生まれてくる。解像度としては最近のインクジェットプリンタのほうがレーザープリンタよりも（SX-WINDOWで扱う限り）高くなっているのだが，印字品質はレーザープリンタに軍配が上がる。カラーとモノクロを組み合わせれば（なんて贅沢），非常に美しい文書を作成することもできるだろう。

やろうと思えばレーザープリンタのコード印字と組み合わせることも不可能ではないのだが……。

グラフィックを扱うには

たとえば，512×512ドットのグラフィックを文書に張り込んで印刷を行おうとすると，すぐにメモリが足りなくなる。

この状況は張り込むときにどういう形式を選ぶかで多少変わってくる。PIC形式でコピーしてペーストすればプリントアウトにやや時間がかかるが，メモリに少しだけ融通がきくようになる。全然プリントアウトできないよりは，少々時間がかかるくらいのほうがまだ我慢できる。

モノクロ印刷なら元絵の段階であらかじめ加工しておくことでPICの圧縮率をかなり上げることもできる。

さらにいいのは，PAT4（パターンエディタのファイル形式）にまであらかじめ落としておくことである。写真などの階調の多いものでは難しいが，図版風なものなどではキャンバス.Xで変換するだけ，またはG2T.Xなどで変換したものを使うことでかなり効率を上げることができる。

XDTPを使えば，それほど極端なメモリ消費をしなくても印刷できるのだが，そっちはテキストのほうの扱いがいまひとつである。ここで2度打ちが重要になってくることはわかるだろう。

最初に簡単なレイアウトを作っておけばあとはそれに従って作業を進めるだけだ。これまでは，シャーペンに限らずSX-WINDOWのカラー印刷はかなり変な仕上がりだったのであまり問題にされていなかったのだが，ワープロパックver.2.0ではかなり高性能なカラー印刷ドライバが付属しているということなのでシャーペンでグラフィック印刷を行うことも増えてくるだろう。

現状ではグラフィックを使うとプリントアウトが控えめにいっても「非常に」遅くなる。これはシャーペンワープロパックver.2.0の発売で解決されるはずである。

まともにグラフィック印刷までできるようになれば，シャーペンの実用度はいっそう上がることになる。できあがる文書のクオリティ限界も一気に上昇する。これで「SX-WINDOWでDTP」という言葉も空々しいものには聞こえなくなることだろう。

ワープロパックは事実上，SX-WINDOW作業の要となるシャーペンのバージョンアップであり，Easydraw，書家万流などにもまして欠くことのできないツールになりつつあるようだ。

［特集］

Now Printing

解像度変換へのアプローチ

輪郭保護拡大処理

Kikuchi Isao 菊地 功

パソコンの画面とプリンタでは解像度が異なる
そこで画像の拡大処理が必要になってくる
プリンタの解像度を損なわず原画を美しく拡大する方法を考えよう

ここ最近のカラープリンタの動向には目を見張るものがあります。各社から次々と発表される新製品のサイクルは，短いといわれる（X680x0を除く）パソコン本体のそれをも凌駕してしまっています。360dpiは当たり前，600dpiや720dpiといった，化け物級のプリンタまでもが，手軽にとはいかないものの，ちょっと頑張れば手の届く値段まで下がってきています。

ただ，ご存じのようにWindowsとMacのドライバは当たり前のようについていても，X680x0で使えるドライバがついたプリンタなど，ただのひとつもありません。「MATIERの対応待ちだな」と考えている人も多いでしょう。しかし，それを待っていたのでは，きっと対応したころには次のもっといいプリンタが出ていることでしょう。

やはり，ないものは自分で作ってしまうというのがいちばんです。下手にタコなドライバが付属していれば，多少不満があったところで，それで我慢してしまうでしょうから，ドライバがないというのは，かえって恵まれた環境なのかもしれないと思うことにしましょう。それにしても，最近のプリンタには制御コードが解説されていないものがあまりに多いのは嘆かわしいことです。

さて，プリンタに出力するには，いくつかの手順を踏まなくてはなりません。そうそう，ここでの出力というのは，グラフィックの出力の話です。テキストならばいくらでも方法はありますからね。

たとえば，512×512ドットのグラフィックを出力するとしましょう。仮に720dpiのプリンタで出力したならば，わずか2cm四方の正方形になってしまいますよね。つまり，これを適当な大きさで正しいアスペクト比で出力するには，まず拡大という処理を行わなくてはなりません。

次に，現在主流のカラーインクジェットプリンタは，YMCの3色，もしくはそれに黒を加えた4色のインクの混合によって表現されますので，それに合わせて減色することになるでしょう。

最後に，ここまでの手順で得られたデータを，それぞれのプリンタのフォーマットにしたがって流し込んでやります。このフォーマットについては，プリンタのマニュアルで説明されていることもあれば，メーカーに問い合わせなければならない場合もありますが，減色以降は瀧氏の記事に譲るとして，ここでは拡大について考えてみましょう。

拡大いろいろ

ひと言で拡大といっても，いろいろな方法が考えられます。簡単なところを挙げてみましょう。

・単純拡大

1ドットを正方形（長方形）とみなして，そのまま拡大します(図1)。一番簡単ですが，モザイクのようになり，美しくありません。

・線形補間拡大

隣り合うドット間の輝度を線形補間して拡大します(図2)。滑らかになりますが，全体的にぼけたような画像となり，輪郭が曖昧になります。

・輪郭保存線形補間拡大

輝度の近いドット間だけを線形補間して拡大します（図3）。輪郭は保存されますが，斜めの輪郭は階段状になってしまいます。

図1 単純拡大



図2 線形補間拡大



ほかにも加重平均拡大というものもありますが，この方法は任意の倍率に拡大するのが難しいのでおいておきました。また，画像を周波数成分に分解して構成し直すフーリエ変換や，離散コサイン変換などを利用した方法も考えられますが，私の能力では労力に見合うだけの結果が得られそうになかったので，上に挙げた輪郭保存線形補間拡大を拡張した直接的な方法を考えてみます。

輪郭補正線形補間拡大

まずは輪郭保存線形補間拡大を考えてみましょう。いま，図4のような2×2のドット列を考えます(点線内が描画領域)。ここで，もしこの4ドットすべてが近い輝度であった場合（図左）は4点を四方として輝度を線形補間し，ドット間の輝度を決めます。4隅をそれぞれのドットのカラーコードとしたグラデーションと考えればいいでしょう。

4ドット中3ドットおよび2ドットが近い輝度であった場合も，それぞれ図中および図右のようになります（対角の2ドットが類似色であった場合はいまは考えないことにします)。グラフィック全体の隣接するすべての2×2ドットに対してこの処理を施すことで，とりあえず輪郭保存線形補間拡大はできました。

さて，この方法では，斜めの輪郭は図5左のように，階段状になってしまいます。これを綺麗な輪郭にするために，図の点線のように補正します。一見めんどう臭そうに見えますが，実際は図4中で示したパターンに図5右のように三角形のパーツをはめ込むだけですんでしまいます。

図3　輪郭保存線形補間拡大



ただし，これだけでは図6左のような斜め線には対応できませんので，これは別に考えてやらなければなりません。図の点線のように補正するのは，図5の補正に加えて，図6右のようにパーツをはめ込んでやればいいのですが，ここでちょっとやっかいな状況に直面します。I_1とI_4が全然違えばなんの問題もないのですが，互いの対角同士が類似色であった場合，どちらがフォアグラウンドカラーなのか，この2×2ドットからでは判断できません。そこで，この4ドットのそれぞれの8近傍の色を調べて，周囲に類似色が多いほうをバックグラウンドとすることにします。込み入った部分では間違えることもありますが，それはご愛敬ということで……。

さて，ここまでの方法で図7左のような輪郭を補正すると，図7中のようになります。これでもそれなりにドットが目立たなくなったと思うのですが，ここはもうちょっと欲張って，もう一段階傾斜を持たせて図7右まで頑張ってみましょう。

そのためには，いままでの2×2パターンだけではなく，図8左のような場合には点線で示したドットまで足を伸ばさなければなりません。ただし，このときEとFは

図4　輪郭保存線形補間拡大のようす

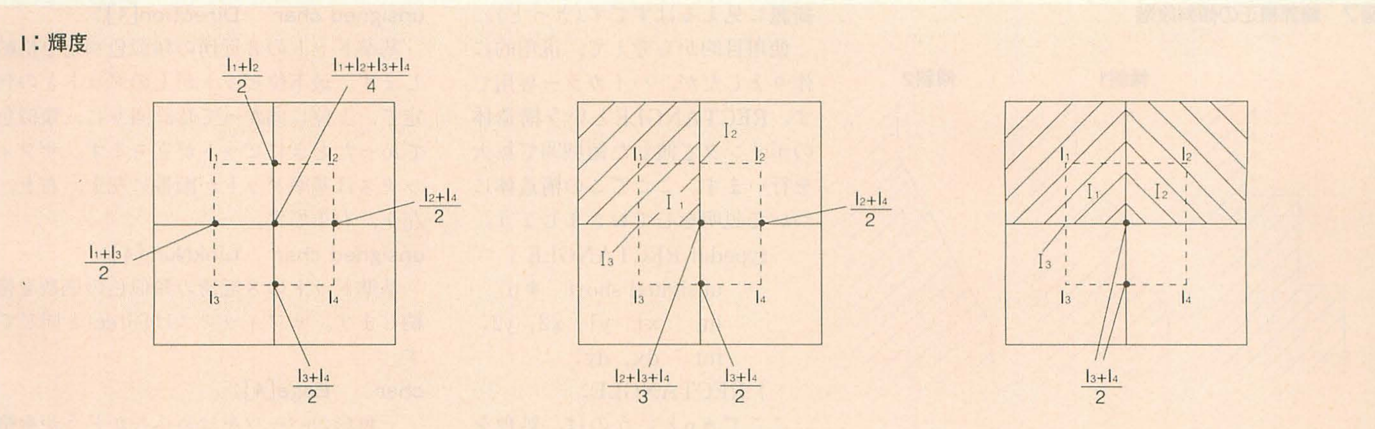


図5　斜め輪郭補正

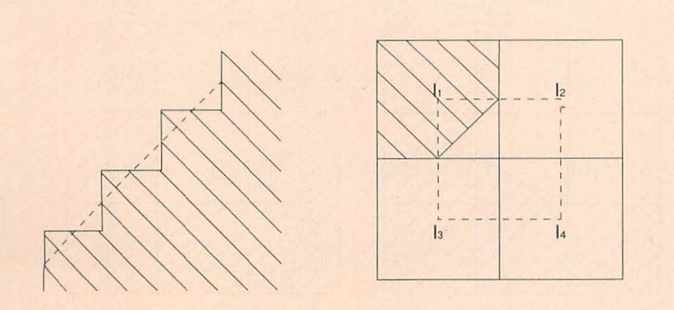


図6　斜め線補正

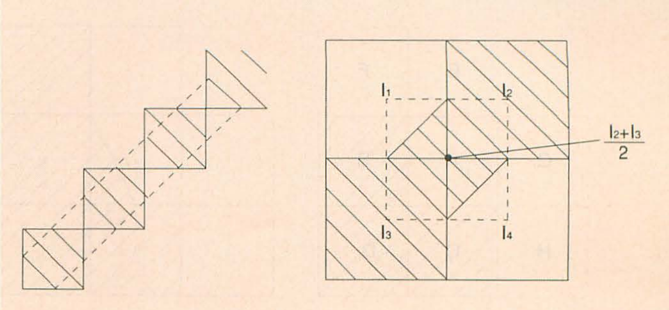


図9　輪郭補正線形補間拡大

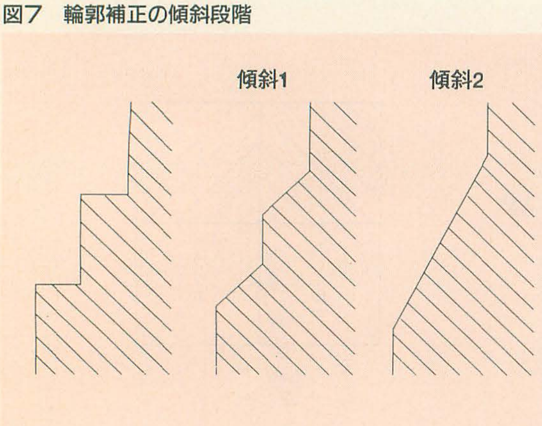


Bと，GとHはCと類似色であるかどうかを調べるだけですみますので，やはりベースは2×2で考えていくことができます。これらの4ドットのうち，図8右の3つの場合だけを例外的に扱うようにします。真ん中の2つは新しい傾斜を加えるものですが，一番右はエッジを尖らせるためのものです。こうしておかないと，単なる四角形の場合でも，角が落ちてしまいますから。

さて，ここでまた問題が出てきました。真ん中の2つの場合には，三角形のパーツが点線で示した描画領域をはみ出してしまいます。はみ出した領域は，たとえば左から2番目の図の場合，ひとつ上を描画するとき(EFABが基準)に処理してしまうことも考えられますが，GとHを判断するためのCが2×2の外にあり，ちょっとやっかいです。そこで，この場合に限って，描画領域を超えてパーツを描画し，その領域をフラグで管理することにしました。

あと，線にも対応させるために，具体的には図8でDが類似色でない場合の処理を加えます。この場合は，さらに右と下のドットとの判断も必要になってきますが，基本的には同じことなので，説明は省略します。

図7　輪郭補正の傾斜段階



プログラム

これを関数にしたのがリスト1，ヘッダファイルがリスト2，その関数を使って画面右下の100×100を拡大するためのプログラムがリスト3，そのプログラムで拡大したのが図9です。画面で見てしまうと，「な〜んか，いまいち」と思うかもしれませんが，ドットの細かいプリンタに出力すれば，綺麗に見えるはずです(きっと)。

使用目的から考えて，汎用的に作りましたが，ハイカラー専用です。RECTANGLEという構造体のポインタで渡した領域間で拡大を行います。ここでこの構造体について説明をしておきましょう。

```
typedef RECTANGLE {
    unsigned short  *p;
    int   x1, y1, x2, y2;
    int   dx, dy;
} RECTANGLE;
```

ここで＊pというのは，処理を始めたい点のアドレスではなく，確保されているメモリの先頭アドレスを示し，実際の処理開始アドレスはp＋x1＋y1＊dxとなります。

この式からもわかるように，dxとdyは処理領域の大きさではなく，確保されているメモリの大きさを示します(ドット，つまり16ビット単位)。つまり，画像の任意の矩形領域を任意の位置に拡大できるわけです。プリンタの画素数を考えれば，数千ドットへの拡大になるわけですから，分割拡大は必須ですよね。

拡大関数はzoom()で，第1引数で与えられたRECTANGLE構造体のポインタから，第2引数で与えられたポインタへ拡大します。基本的にいままで説明した処理を行っているだけなのですが，そのままプログラムしたのでは意味もなくリストが長くなってしまい，鬼のように面倒なことになってしまうので，配列のサフィックスや値を配列に放り込んでおいて，それをforループで参照するということをやっています。ですから一見してなにをやっているのかわからないかもしれませんが，そのへんはご容赦ください。

とりあえず，いくつかの変数と関数の意味を説明しておきましょう。

`unsigned char　RGB[16][3];`

描画する領域を4分割し，それぞれの4隅の色を格納します(図10)。

`unsigned char　Direction[4];`

基準ドットの8近傍の類似色判定を格納します。最下位ビットが上のドットとの判定で，上位に向かって時計回りに，類似色であったときにビットが立ちます。サフィックスは基準ドットが順番に左上，右上，左下，右下です。

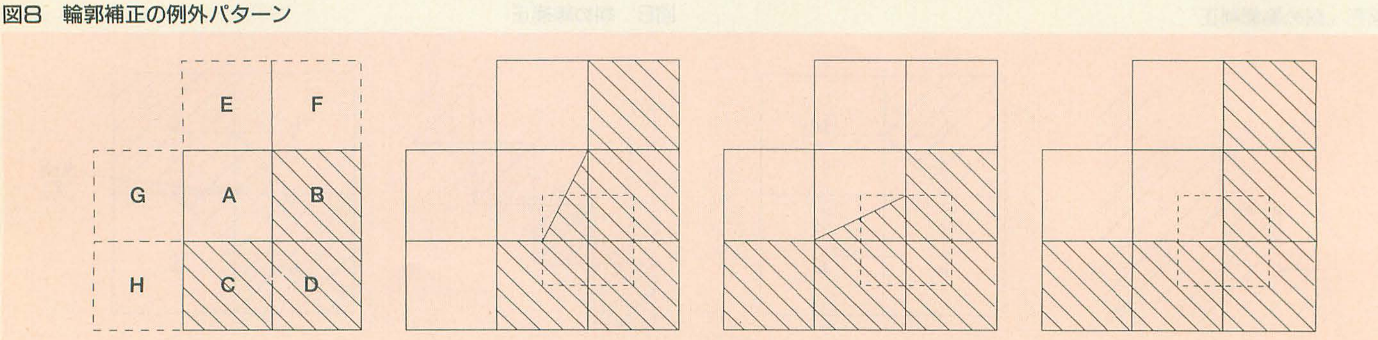
`unsigned char　LinkNum[4];`

基準ドットの8近傍の類似色の個数を格納します。サフィックスはDirectと同じです。

`char　Edge[4];`

三角形のパーツをはめ込むかどうかを格

図8　輪郭補正の例外パターン



納します。0のときはなし，1は傾斜1,2と3は傾斜2です。これもサフィックスはDirectと同じです。

void　PreProcess();

基準の2×2パターンに対して，類似色フラグDirectおよび類似色個数LinkNumを求めます。

void　PutRectangle();

RGB変数を元に，矩形を線形補間して描画します。このとき，最下位ビットの未処理フラグがすでに0になっているところは描画しません。

void　PutEdge();

傾斜2のパーツを描画します。こちらは未処理フラグを無視します。

最初にdefineで定義してあるDIFF_R，DIFF_G,DIFF_Bは，それぞれRGBの輝度がそれ以下であれば，類似色と見なすようになっています。画像によっては，この値を調整する必要があるかもしれません。アンチエイリアスなしのアニメ調の絵ならば，もっと小さくしてもいいでしょう。

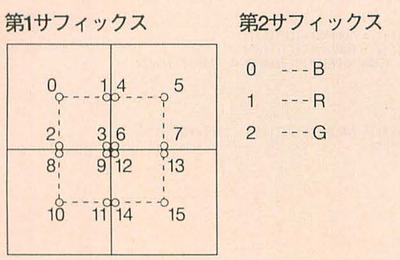
図9をよ〜く見ると，輪郭補正した斜め線がちょっとふらふらしているのがわかるかと思います。これは，拡大時に1ドットを整数ドットに対応させているためで，たとえば，4.3倍に拡大する場合，あるドットは4倍に，あるドットは5倍に拡大されます。ここでまず一次の誤差が生じます。さらに，拡大されたドット内で斜め線を引くのに二次の誤差が生じます。学生時代に，実験中に「誤差が狂う」という発言をして，「まさにごさごさ」と突っ込まれたやつがいましたが，そういうことです。まあ，多少ふらついたところで，プリンタヘッドの精度を考えれば，たいした問題ではないでしょう。

毎度のことで申しわけないのですが，あとはリストを見て理解してください。今月のリストは，ちょっとごちゃごちゃしてるけど，難しくはないよ，ね，ね？

暑いっすー

部屋は暑いし，新型は気になるし，母校はベスト8で負けるしで，今月はなかなか原稿がはかどらなかったなぁ。いや，(U)氏には申し訳ないと思ってるんですよ（ほんとか？）。とりあえず，この号が出るころには涼しくなってくれてるといいなぁ。MOの調子も悪かったし。もっとも，6メートルぎりぎりのSCSIケーブルが，暑くて熱抵抗が大きくなってたのが原因だったんだけど。

図10　配列変数RGBの内容



リスト1

```
/* プリンタ出力用輪郭補正線形補間拡大ルーチン        (c)Isawo-Kikuchi          */

typedef     struct RECTANGLE {
    unsigned short *p;       /* 先頭アドレス */
    int     x1, y1, x2, y2; /* 領域座標 */
    int     dx, dy;          /* メモリサイズ */
} RECTANGLE;

extern void         zoom( RECTANGLE *, RECTANGLE * );
```

リスト2

```
/* プリンタ出力用輪郭補正線形補間拡大ルーチン        (c)Isawo-Kikuchi          */

#include    <stdlib.h>
#include    <stdio.h>

#define     DIFF_R  4
#define     DIFF_G  4
#define     DIFF_B  4

typedef     struct RECTANGLE {
    unsigned short *p;       /* 先頭アドレス */
    int     x1, y1, x2, y2; /* 領域座標 */
    int     dx, dy;          /* メモリサイズ */
} RECTANGLE;

void        zoom( RECTANGLE *, RECTANGLE * );

static void PreProcess( unsigned short *, int, int, int, int );
static void PutRectangle( unsigned short *, int, int, int, int, int, int );
static void PutEdge( unsigned short *, int, int, int, int, int, int );

static unsigned char        RGB[16][3];
static unsigned char        Direction[4];
static unsigned char        LinkNum[4];
static char         Edge[4];
static unsigned char        N0[4][7] = {     /* 縦横比較用 */
    0x04, 0, 5, 1, 4, 3, 6,         /* 左上・右上 */
    0x10, 5,15, 7,13, 6,12,         /* 右上・右下 */
    0x01,10, 0, 8, 2, 9, 3,         /* 左下・左上 */
    0x40,15,10,14,11,12, 9 /* 右下・左下 */
};
static unsigned char        N1[4] = {        /* 4点比較用 */
    0x04|0x08|0x10, 0x10|0x20|0x40, 0x01|0x02|0x04, 0x40|0x80|0x01
};
static unsigned char        N2[4][12] = {    /* 3点比較用 */
    0x04|0x10, 1, 0x04, 0x04|0x08, 2, 0x10, 0x08|0x10, 0, 5,10,15,12,
    0x10|0x40, 0, 0x40, 0x20|0x40, 3, 0x10, 0x10|0x20, 0, 5,15,10, 9,
    0x01|0x04, 3, 0x04, 0x02|0x04, 0, 0x01, 0x01|0x02, 0,10,15, 5, 6,
    0x01|0x40, 2, 0x40, 0x40|0x80, 1, 0x01, 0x01|0x80, 5,10,15, 0, 3
};
static unsigned char        N3[4][14] = {
  2, 0x08|0x80, 0x08|0x40, 0x10|0x80,12,12, 3, 3, 0x02|0x20, 0x04|0x20, 0x02|0x10, 9, 9, 6,
  0, 0x02|0x20, 0x02|0x40, 0x01|0x20, 9, 6, 9, 2, 0x08|0x80, 0x08|0x40, 0x10|0x80, 3,12, 3,
  3, 0x02|0x20, 0x04|0x20, 0x02|0x10, 6, 9, 6, 1, 0x08|0x80, 0x04|0x80, 0x01|0x08,12, 3,12,
  1, 0x08|0x80, 0x04|0x80, 0x01|0x08, 9, 9, 6, 0, 0x02|0x20, 0x02|0x40, 0x01|0x20,12,12, 3
};

void        zoom( RECTANGLE *inrect, RECTANGLE *outrect )
{
    int     x, y, i, j, k, flag;
    unsigned short *p1, *p2;
    int     dx1, dy1, dx2, dy2, x0, y0, x1, y1, x2, y2, x3, y3, x4, y4;

    dx1 = inrect->x2-inrect->x1+1;
    dy1 = inrect->y2-inrect->y1+1;
    dx2 = outrect->x2-outrect->x1+1;
    dy2 = outrect->y2-outrect->y1+1;
    p1 = inrect->p;
    p2 = outrect->p;
    p2 += outrect->x1+outrect->y1*outrect->dx;
    for( y=0; y<dy2; y++ ) for( x=0; x<dx2; x++ ) p2[x+y*outrect->dx] |= 1;/*未処理フラグ*/
    for( y=y3=y4=0; y<=dy1; y++ ){
        y0 = y2;
        y1 = y3;
        y2 = y4;
        y3 = (2*y+1)*dy2/dy1/2;
        y4 = (y+1)*dy2/dy1;
        for( x=x3=x4=0; x<=dx1; x++ ){
            x0 = x2;
            x1 = x3;
            x2 = x4;
            x3 = (2*x+1)*dx2/dx1/2;
            x4 = (x+1)*dx2/dx1;
            PreProcess( p1, x+inrect->x1-1, y+inrect->y1-1, inrect->dx, inrect->dy );
            for( i=0; i<4; i++ ){
                Edge[i] = 0;
                if( Direction[i]&N0[i][0] ){        /* 縦横が類似色 */
                    for( j=0; j<3; j++ ){
                        RGB[N0[i][3]][j] = RGB[N0[i][4]][j] =
                        (RGB[N0[i][1]][j]+RGB[N0[i][2]][j])/2;
                    }
                } else {
                    for( j=0; j<3; j++ ){
                        RGB[N0[i][3]][j] = RGB[N0[i][1]][j];
                        RGB[N0[i][4]][j] = RGB[N0[i][2]][j];
                    }
                }
            }
            for( i=0; i<4; i++ )
                if( (Direction[i]&N1[i])==N1[i] ) break;
            if( i<4 ){      /* 2×2が類似色 */
                for( j=0; j<3; j++ ){
                    RGB[3][j] = RGB[6][j] = RGB[9][j] = RGB[12][j] =
                    (RGB[0][j]+RGB[5][j]+RGB[10][j]+RGB[15][j]+2)/4;
                }
            } else {
                for( i=0; i<4; i++ ){
                    /* 3点が類似色 */
                    if( (Direction[i]&N2[i][0])==N2[i][0] ){
                        for( j=0; j<3; j++ )
                            RGB[3][j] = RGB[6][j] = RGB[9][j] = RGB[12][j] =
                            (RGB[N2[i][7]][j]+RGB[N2[i][8]][j]+RGB[N2[i][9]][j]+1)/3;
                        flag = 0;
                        if( Direction[N2[i][1]]&N2[i][2] )
                            flag |= 1;
                        if( Direction[N2[i][4]]&N2[i][5] )
                            flag |= 4;
                        if( flag==0 ) Edge[3-i] = 1;
                        else if( flag==5 ){
                            for( j=0; j<3; j++ )
                                RGB[N2[i][11]][j] = RGB[N2[i][10]][j];
                        } else {
                            if( (Direction[N2[i][flag]]&N2[i][flag+2])==N2[i][flag+2] ){
                                Edge[3-i] = 1;
                            } else {
                                Edge[3-i] = (flag==1)?2:3;
                            }
                        }
                        break;
                    }
                }
                if( i==4 ){
                    for( i=0; i<4; i++ ) for( j=0; j<3; j++ ) RGB[i*3+3][j] = RGB[i*5][j];
                    for( i=0; i<4; i++ ){
                        /* 縦横2点が類似色 */
                        if( (Direction[i]&N0[i][0])==N0[i][0] ){
```

```
                            for( j=0; j<3; j++ ){
                                RGB[N0[i][5]][j] = RGB[N0[i][3]][j];
                                RGB[N0[i][6]][j] = RGB[N0[i][4]][j];
                            }
                            if( (Direction[3-i]&N0[3-i][0])!=N0[3-i][0] ){
                                for( j=1; j<4; j++ ){
                                    if( (Direction[N3[i][0]]&N3[i][j])==N3[i][j] ){
                                        Edge[N3[i][0]] = -1;
                                        k = N3[i][j+3];
                                        for( j=0; j<3; j++ )
                                            RGB[N3[i][0]*3+3][j] = RGB[k][j];
                                        break;
                                    }
                                }
                                for( j=1; j<4; j++ ){
                                    if( (Direction[N3[i][7]]&N3[i][j+7])==N3[i][j+7] ){
                                        Edge[N3[i][7]] = -1;
                                        k = N3[i][j+10];
                                        for( j=0; j<3; j++ )
                                            RGB[N3[i][7]*3+3][j] = RGB[k][j];
                                        break;
                                    }
                                }
                            }
                        }
                    }
                    flag = 0;
                    if( Direction[0]&0x08 ) flag |= 1;      /* 対角が類似色 */
                    if( Direction[1]&0x20 ) flag |= 2;
                    if( flag==3 )
                        if( LinkNum[0]+LinkNum[3]>LinkNum[1]+LinkNum[2] )
                            flag = 2; else flag = 1;
                    if( flag==1 ){
                        for( j=0; j<3; j++ ){
                            RGB[3][j] = RGB[6][j] = RGB[9][j] = RGB[12][j] =
                            (RGB[0][j]+RGB[15][j])/2;
                            Edge[1] = Edge[2] = 1;
                        }
                        for( i=0; i<3; i++ ){
                            flag = 0;
                            if( Direction[N2[i][1]]&N2[i][2] )
                                flag |= 1;
                            if( Direction[N2[i][4]]&N2[i][5] )
                                flag |= 4;
                            if( flag==0 || flag==5 ) Edge[3-i] = 1;
                            else {
                                if( (Direction[N2[i][flag]]&N2[i][flag+2])==N2[i][flag+2]){
                                    Edge[3-i] = 1;
                                } else {
                                    Edge[3-i] = (flag==1)?2:3;
                                }
                            }
                        }
                    } else if( flag==2 ){
                        for( j=0; j<3; j++ ){
                            RGB[3][j] = RGB[6][j] = RGB[9][j] = RGB[12][j] =
                            (RGB[5][j]+RGB[10][j])/2;
                            Edge[0] = Edge[3] = 1;
                        }
                        for( i=0; i<4; i+=3 ){
                            flag = 0;
                            if( Direction[N2[i][1]]&N2[i][2] )
                                flag |= 1;
                            if( Direction[N2[i][4]]&N2[i][5] )
                                flag |= 4;
                            if( flag==0 || flag==5 ) Edge[3-i] = 1;
                            else {
                                if( (Direction[N2[i][flag]]&N2[i][flag+2])==N2[i][flag+2]){
                                    Edge[3-i] = 1;
                                } else {
                                    Edge[3-i] = (flag==1)?2:3;
                                }
                            }
                        }
                    }
                }
            }
            if( x>0 && y>0 ){
                PutRectangle( p2, x1, y1, x2, y2, outrect->dx, 0 );
                if( Edge[0]==2 ) PutEdge( p2, x0, y1, x1, y2, outrect->dx, 0 );
                if( Edge[0]==3 ) PutEdge( p2, x1, y0, x2, y1, outrect->dx, 0 );
            }
            if( x<dx1 && y>0 ){
                PutRectangle( p2, x2, y1, x3, y2, outrect->dx, 1 );
                if( Edge[1]==2 ) PutEdge( p2, x3, y1, x4, y2, outrect->dx, 1 );
                if( Edge[1]==3 ) PutEdge( p2, x2, y0, x3, y1, outrect->dx, 1 );
            }
            if( x>0 && y<dy1 ){
                PutRectangle( p2, x1, y2, x2, y3, outrect->dx, 2 );
                if( Edge[2]==2 ) PutEdge( p2, x0, y2, x1, y3, outrect->dx, 2 );
                if( Edge[2]==3 ) PutEdge( p2, x1, y3, x2, y4, outrect->dx, 2 );
            }
            if( x<dx1 && y<dy1 ){
                PutRectangle( p2, x2, y2, x3, y3, outrect->dx, 3 );
                if( Edge[3]==2 ) PutEdge( p2, x3, y2, x4, y3, outrect->dx, 3 );
                if( Edge[3]==3 ) PutEdge( p2, x2, y3, x3, y4, outrect->dx, 3 );
            }
        }
    }
}

static void        PreProcess( unsigned short *p, int x0, int y0, int dx, int dy )
{
    int    x, y, x1, y1, x2, y2, i, j, r0, g0, b0, r, g, b;
    unsigned short col;

    for( y=0; y<2; y++ ){
        y1 = y0+y;
        if( y1<0 ) y1 = 0; else if( y1>=dy ) y1 = dy-1;
        for( x=0; x<2; x++ ){
            j = x+y*2;
            x1 = x0+x;
            if( x1<0 ) x1 = 0; else if( x1>=dx ) x1 = dx-1;
            Direction[j] = LinkNum[j] = 0;
            for( i=0; i<8; i++ ){
                col = p[x1+y1*dx];
                RGB[j*5][0] = b0 = (col>>=1)&0x1f;
                RGB[j*5][1] = r0 = (col>>=5)&0x1f;
                RGB[j*5][2] = g0 = col>>5;
                x2 = x1; y2 = y1;
                if( i==0 || i==1 || i==7 ){         /* 上成分 */
                    y2 = y0+y-1;
                    if( y2<0 ) y2 = 0;
                }
                if( i>=1 && i<=3 ){                 /* 右成分 */
                    x2 = x0+x+1;
                    if( x2>=dx ) x2 = dx-1;
                }
                if( i>=3 && i<=5 ){                 /* 下成分 */
                    y2 = y0+y+1;
                    if( y2>=dy ) y2 = dy-1;
                }
                if( i>=5 && i<=7 ){                 /* 左成分 */
                    x2 = x0+x-1;
                    if( x2<0 ) x2 = 0;
                }
                col = p[x2+y2*dx];
                b = (col>>=1)&0x1f;
                r = (col>>=5)&0x1f;
                g = col>>5;
                if( (r0>r?r0-r:r-r0)<=DIFF_R &&
                    (g0>g?g0-g:g-g0)<=DIFF_G &&
                    (b0>b?b0-b:b-b0)<=DIFF_B ){
                    Direction[j] |= 1<<i;
                    LinkNum[j]++;
                }
            }
        }
    }
}

static char        N4[4][6][4] = {
    0, 2, 0, 0,  2, 2,-2, 0,  2, 2,-4, 0,  1, 2,-1, 0,  0, 4, 0,-4,  0, 2, 0,-1,
    0, 2, 0, 0,  0, 0, 0, 2,  0, 0, 0, 4,  0, 1, 0, 1, -2, 2, 4, 0,  0, 2, 1, 0,
    0, 2, 0, 0,  0, 2, 2, 0, -2, 2, 4, 0,  0, 2, 1, 0,  0, 0, 0, 4,  0, 1, 0, 1,
    0, 2, 0, 0,  0, 2, 0,-2,  0, 4, 0,-4,  0, 2, 0,-1,  2, 2,-4, 0,  1, 2,-1, 0
};
static unsigned char      N5[4][4] = {
     1, 2, 4, 8,
     4, 7, 1,13,
     8,11, 2,14,
    13,14, 7,11
};

static void PutRectangle(unsigned short *p, int x1, int y1, int x2, int y2, int dx0, int i)
{
    int    x, y, dx, dy, j;
    int    rgb[3][3];
    unsigned short col;
    int    sx, ex, dx1, dx2;

    dx = x2-x1;
    dy = y2-y1;
    if( Edge[i] ){
        col = ((((RGB[i*5][2]<<5)|RGB[i*5][1])<<5)|RGB[i*5][0])<<1;
        if( Edge[i]>0 ){
            for( j=0; j<3; j++ ){
                RGB[N5[i][0]][j] = RGB[N5[i][2]][j];
                RGB[N5[i][1]][j] = RGB[N5[i][3]][j];
                RGB[i*5][j] = (RGB[N5[i][0]][j]+RGB[N5[i][0]][j])/2;
            }
        } else {
            Edge[i] = -Edge[i];
            for( j=0; j<3; j++ )
                RGB[N5[i][0]][j] = RGB[N5[i][1]][j] = RGB[i*5][j] =
                    RGB[i*3+3][j];
        }
    }
    sx = N4[i][Edge[i]][0];
    ex = N4[i][Edge[i]][1];
    dx1 = N4[i][Edge[i]][2];
    dx2 = N4[i][Edge[i]][3];
    for( y=0; y<dy; y++ ){
        for( j=0; j<3; j++ ){
            rgb[0][j] = RGB[i*4  ][j]+(RGB[i*4+2][j]-RGB[i*4  ][j])*y/dy;
            rgb[1][j] = RGB[i*4+1][j]+(RGB[i*4+3][j]-RGB[i*4+1][j])*y/dy;
        }
        for( x=0; x<(sx*dx+dx1*dx*y/dy)/2; x++ ){
            if( x>=dx ) break;
            p[x+x1+(y+y1)*dx0] = col;
        }
        for( ; x<(ex*dx+dx2*dx*y/dy)/2; x++ ){
            if( x>=dx ) break;
            if( p[x+x1+(y+y1)*dx0]&1 ){
                for( j=0; j<3; j++ ) rgb[2][j] = rgb[0][j]+(rgb[1][j]-rgb[0][j])*x/dx;
                p[x+x1+(y+y1)*dx0] = ((((rgb[2][2]<<5)|rgb[2][1])<<5)|rgb[2][0])<<1;
            }
        }
        for( ; x<dx; x++ ) p[x+x1+(y+y1)*dx0] = col;
    }
}

static void PutEdge( unsigned short *p, int x1, int y1, int x2, int y2, int dx0, int i)
{
    int    x, y, dx, dy;
    unsigned short col;
    int    sx, ex, dx1, dx2;

    dx = x2-x1;
    dy = y2-y1;
    sx = N4[i][Edge[i]+2][0];
    ex = N4[i][Edge[i]+2][1];
    dx1 = N4[i][Edge[i]+2][2];
    dx2 = N4[i][Edge[i]+2][3];
    col = ((((RGB[i][2]<<5)|RGB[i][1])<<5)|RGB[i][0])<<1;
    for( y=0; y<dy; y++ ){
        for( x=0; x<(sx*dx+dx1*dx*y/dy)/2; x++ ){
            if( x>=dx ) break;
            p[x+x1+(y+y1)*dx0] = col;
        }
        for( ; x<(ex*dx+dx2*dx*y/dy)/2; x++ ){
            if( x>=dx ) break;
        }
        for( ; x<dx; x++ ) p[x+x1+(y+y1)*dx0] = col;
    }
}
```

リスト3

```
#include    <stdlib.h>
#include    <stdio.h>
#include    <doslib.h>
#include    <iocslib.h>
#include    "zoom.h"

RECTANGLE   inrect, outrect;

void        main()
{
    inrect.p = outrect.p = 0xC00000;
    inrect.x1 = 512-100;
    inrect.y1 = 512-100;
    inrect.x2 = inrect.y2 = 511;
    outrect.x1 = outrect.y1 = 0;
    outrect.x2 = outrect.y2 = 511;
    inrect.dx = inrect.dy = outrect.dx = outrect.dy = 512;

    SUPER( 0 );
    zoom( &inrect, &outrect );
}
```

「特集」

New Printing

印刷処理の基本

減色処理と印刷いろいろ

Taki Yasushi 瀧 康史

印刷処理の基本になる事項についてまとめて考えてみよう
ここでは画像のプリンタ印字での基本になる
減色処理とモノクロ化を中心に印刷プログラムの作り方を考える

絵描きでない人が，1枚のCGを印刷するというのは，ある意味特別なことです。強いて近い感情をあげるなら，ウィンドウの壁紙に使うことでしょうか。MOやHDの中に，CGをしまっておくのではなく，自分の目の届くところへ置きたいという気持ち。つまりそれは，それだけそのCGを気にいったということに間違いありません。

気にいったCGだからこそ，できるだけ綺麗に見たい。だから，できるだけ見栄えがよくなるように加工をしてみます。ウィンドウの背景に敷きたい場合なら，せいぜい，アスペクト比を直したり，ちょうどよい大きさに拡縮したり，減色したりするぐらいでしょう。ほら，やったことがありませんか？　気にいった絵をまとめて，grroot.xのために，512×512のPICにしたりとか。

印刷の場合だとこだわりはさらに倍増します。私の場合，タブレットなどを使って，デッサンが気にいらないところを修正することだってあります。他人の絵なのにね。

ここまですることはそうはないですが，それでもなんらかのフィルタなどを通していろんな処理を加えたりします。たとえば，テレビからの取り込み画像ならばメディアンフィルタなどのノイズフィルタや，インタレスフィルタ。イメージスキャナからの取り込みなら，メディアンフィルタ1回程度。手描きでも完成度が高ければそのまま印刷してみますが，見た目に気にいらないところがあるとやっぱり直します。512×512のハイカラー画像ならばマッハバンドイレーサ，ジャギーイレーサなど。パレット付き16色画像ならば一度フルカラー画像に変換し，タイルフィルタを通したり，ジャギーイレーサを通したりします。

ほしいのははっと見て，壁にかかった絵に感動する瞬間。絵が描ければそれを印刷するのでしょうが，私のように絵心に乏しいものは，こうしていろんな人の絵をいただいていたりするのです。

印刷の手順

菊地氏の記事にも書かれておりますが，グラフィックを印刷するためには，手順が必要です。まずはプリンタのサイズにあわせるための拡大処理，減色処理です。細かにいうならば，減色処理の手前に，RGBからYCMBへの色座標変換が必要になります。

隣の家ではどんなことをやっているか？ということで，WINDOWSのドライバでいろいろ印刷してみることにしました。

まず拡大処理ですが，これは解像度の低い画像を印刷してみればわかります。512×512のPIC画像は，720dpiのプリンタからみれば，十分小さな画像です。MJ5000Cの印刷サンプルを見る限り，1ドットが四角くなるところをみると，やはりゼロ補間をしているようです。ゼロ補間というのは，拡大して埋める分の画素を，同じ色で割り当てるという方法です。この方法は1ドットが四角くなる分，エッジが綺麗に切断された感じがします。ドット同士の結びつきがほとんどないので，全体的にあっさりした出力結果になります。

拡大処理に直線補間するものもあります。直線補間は，拡大して埋める分の画素を，隣の色と輝度の直線で結び，補間するというものです。これを利用すると，1ドットが四角くならなくなり，隣のドットと滑らかにつながります。しかし，隣のドットと滑らかにつながるということは，1ドットの臨界点がなくなるということであるため，全体的に「ぼかし」を食らったような画像になります。アニメ絵の場合，意識されたエッジがあるので，エリアシングは必要です。

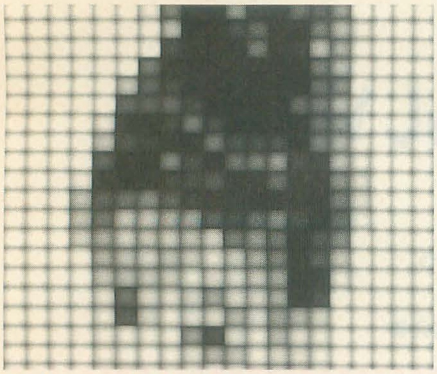
そのため，直線補間は写真などにはわりといいのかもしれませんが，エッジがくっきりしたアニメ絵などにはあまり向かないと，私は思っています。個人的には，写真の場合であっても，直線補間の拡大はあまり好きではありません。

このほか，ベジエラインで輝度差を修復するものなどがありますが，なめらかにつなげればその分，ドット同士がにじみやすくなります。拡大技術はいろいろなことが考えられているようです。菊地氏が今月の記事でいろいろと考えていましたが，私のほうでも考えてみたことを，あとでまとめてみます。

色座標変換は，ディスプレイの表示系であるRGBから，インクの表示系であるYCMBに変換することです。理論的には，ビットを反転するだけでこの処理は終わりです。しかし，これにもいろいろと問題はあります。RGBに比べ，インクのYCMBは，一律の濃度には見えないからです。たとえば，インクはYCMB，つまりイエロー，シアン，マゼンタ，ブラックですが，この4色で「青」を表現するのはかなり難しい。理論上はマゼンタ+シアンですが，素直に合成すると紫っぽくなってしまいます。そのため，結果的にはインバースをするのではなくて，適当なテーブルを作る必要があるようです。最近ではスキャナとセットでカラーグラフを作り，テーブルを自動作成するソフトもあるようです。

減色処理は，誤差分散がかなり有力です。WINDOWS用に用意された各社のプリンタドライバは，ほぼすべて誤差分散を利用しているようです。仕組みを簡単に説明すると，誤差分散は減色時に失われた情報を，進行方向などに振り分ける処理です。Oh!Xでは古くから似たような方法で，桒野式分散がありますが分散の値の計算式が，桒野式のほうはコンピュータに扱いやすい値なので，Oh!Xでは8ビット時代から愛用されてきました。オプティマイズすると計算式が高速化できます。

現在はコンピュータの速度に比べて，プリンタの処理時間のほうがあらゆる意味で



ガウシャンパルスの実行例

遅いため，計算が多少重い誤差分散でも，ほとんど変わらなくなってきました。プリンタのヘッドの移動，セントロ準拠のデータ転送速度など，いろんなところでCPUが待つようです。速度面以外では，振り分け率の違いなどから，桑野式は淡い絵に関して独特の表情を見せます。

誤差分散以外にメジャーなのは，ディザ変換です。ディザにはいろいろありますが，WINDOWSのドライバで使われているのは，ベイヤー法というマトリクスディザのようです。ディザ法はどちらかというと，隣接画素同士の結びつきが減る分，エッジが保存されます。ただ，ベイヤー法のようにディザマトリクスが共通であると，ディザパターンがばれてしまうという欠点があります。ばれたからどうなんだ？　といわれそうですが，サンプルを見ていただければわかるとおり，あまり美しいものではないんですね。淡い色の場合，誤差分散なら，人間の目から見ると，ほぼ無作為にドットが並びますが，ディザでは同じ調子で並びます。見方によってはこれがノイズのように見えるわけです。

ディザパターンは，減色する色数によっても異なりますが，広ければ広いほどディザマトリクスがばれにくくなります。ただし，720dpiという強烈な細かいサイズまでいくと，かなり大きくしてもディザマトリクスの面積が小さくなるため，発見されやすくなってしまいます。離れて見れば，あまり気になりませんが，近くで見るとわりと気になるものですね。このあたりは詳しくあとで話しましょう。

WINDOWSのドライバをこうしていろいろと見てみると，結構いろいろ考えて作られているようです。プリンタとドライバがセットで販売される世の中ですから，ドライバが腐ってると，プリンタも腐っているといわれるわけです。

デバイスドライバはファームと同じようなものですから，いままでROM化されていたファームが，多容量メモリを持つようになったコンピュータ側に移動されたと考えれば済む話です。こうすれば，あとからドライバのバージョンアップができますし，大手ネットワークに行くと，プリンタドライバがサポートされています。

これでとばっちりを食らうのが，マイナーマシンとマイナーOSのユーザーってことですか……。X68000ほどマイナーまでいかなくても，OS/2warpレベルにちょっといくだけで，このことは苦労するようです。ああ，与太話か，これは。

画像サイズが必要なこと

減色処理は次の項でいくつか挙げますが，だいたい誤差分散かディザの大きく分けて2通りでしょう。いい加減さを承知でいってしまえば，減色処理はもはやあまり気にせずとも，ちゃんとしているものを選べば，ほとんど変わらないともいえます。最近のプリンタは高性能ですし，解像度もかなり高くなってきているからです。解像度が高くなれば，その分，人間の目が誤魔化されたり，分散する誤差の数が増えたりしますから。

これと反対に，解像度が減っていくと，画像拡大を真面目に考えなくてはなりません。IO-735X時代に180dpiで印刷した画像と，720dpiを印刷した画像を比べると，1ドットの臨界点がかなりくっきりしてきています。そして1画素に相当する階調度も増えているのです(プリンタドット/画素に影響する)。

WINDOWSのプリンタドライバは，ゼロ補間，直線補間が大半でした。拡大アルゴリズムはこれ以外に，いくつかありますが，どれも一長一短です。印刷する絵柄に応じて，印刷する側が選ばなくてはいけない部分でしょう。なにしろ，拡大という処理は，存在しない情報をいかに誤魔化すか？　ともいい換えることができるほど，無情な処理ですから。

ゼロ補間と直線補間の細かい説明は，本誌1994年10月号の私の記事を参照のこと。これらの知識を応用して，自分でも拡大アルゴリズムを考えてみました。

●ガウシャンパルス補間

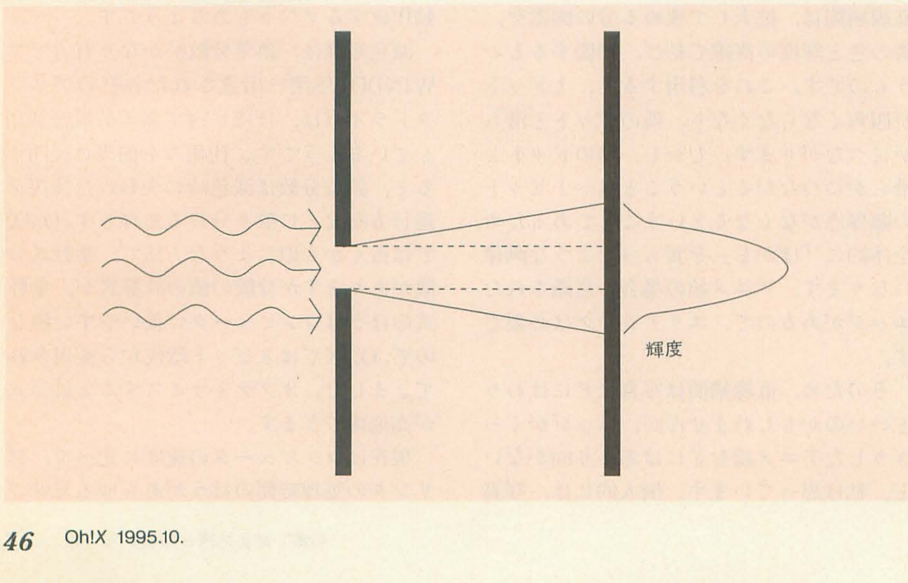
「形が判別できないほど十分小さい1点」を見たとき，人間には●に見えるでしょうか？　■に見えるでしょうか？　仮にその点が四角であったとしても，人間にはにじんで見えます。にじむことによって，1点は丸っぽく見えることになります。また，この「形が判別できない点」が隣接して2つあったとき，連結して見えます。

これがそのまま大きくなったなら，どうなるのだろうか？　ということで，ガウシャンパルス補間という方式を実験してみました。

まず独立した1点を考えます。点であると二元的に定められた長さを取らねばならないため，軸を一元的にわかりやすく考えます。そこで，長さが十分長い1ラインを考えます。1ラインはスリットを通ってきたコヒーレンス光ということで定義し，図1のようにバックスクリーンに映すと，回折しにじんで見えます。隣接して2つの回折があった場合，これらはお互いに干渉しあいます。この法則を応用し，二元的に有限の長さで1ドットを作ります。

画面の拡大率は，回折格子とスクリーンの長さで決まります。色はにじみますが，1ドットは四角くはならずに，むしろ丸くなります。しかし，この式は不確定性理論

図1　ガウシャンパルスの考え方



を応用せねばならないので，真面目に計算すると非常に計算量が増えてしまいます。

なんだか，アダマール行列のウォルシュ関数にどんどん似てきましたが，これを簡略化するために，1ドットの回折率をガウシャンパルス（ガウスの誤差関数）で近似します。これを3元的に処理し，1ドットが周りに与える誤差の影響率を求めていきます。うまく説明できませんが，イメージ的には，スポンジのボコボコが1ドットになるという感覚でしょうか。

よって，1ドットを近似する式は，1元式で，

F(x)=f×e^{-(x^2)/(nτ^2)}

となります。

ここでfは輝度，xは横軸，τは拡大率になります。nは基本的には2なのですが，お互いの画素同士の影響率にもなるので，変えてしまってもよいかもしれません。

y軸に関しても同じ計算をし，x軸分，y軸分からの合計値を求めます。ガウシャンパルスは，nτの間隔で0になるので，n個飛んだ画素には影響を与えません。

ガウシャンパルスは先端が割と尖った関数なので，あまり線が太くなりすぎることもありません。しかし，お互いの色のにじみ具合が悪いため，ベタに模様ができるという不具合もあります。

こうして苦労して求めた計算でも，やっぱり「ない情報」を計算式で作り出したわけですから，本当に最初から大きな画素を持っている絵と比べると，遥かに見劣りしてしまいます。たとえばMJ5000cであるならば，できればPhotoCDの印刷用サイズである，3072×2048(フルカラー：18Mバイト)の画素がほしいところです。もっとも，こんなに大きなサイズの絵を描いてる人は，そうはたくさんいませんが。都築和彦氏のCD-ROM，Illusionに入った画像を印刷してみると，痛切にそれを感じてしまいます。これには，2048×2048ドットの画像が多量に入っているので……。

減色法

もっとも簡単な減色は単純閾値法です。たとえば256階調から2階調に落とすとき，単純に128を閾値にし，以上か未満かでその画素が1か0になるかが決まります。これだと想像どおり，階調画像はめちゃくちゃになってしまいます。

現在出回っているプリンタは，基本的には単色で2値です。YCMB，それぞれのインクでドットを打つか，打たないか。そのために，なんとか階調を出すように考慮した減色をすることになります。

具体的にいくつかの減色法について話していきましょう。

●ディザ（dither）

ディザは簡単にいえば，多階調の各画素にノイズを加えて2値化するというものです。このノイズというものを，どのようにして持ってくるかがディザの方法です。

1）ランダムディザ

もっとも簡単なのは，ランダムディザです。ノイズを階調分のランダム値として利用するのです。たとえば，256階調の画素があったとします。この画素に対してある幅を持った乱数を加算し，その後決められた閾値で切ります。加算された分だけ閾値を高めにしないと，コントラストが上がってしまうので注意が必要です。ここで，乱数の幅，閾値は適当に決めます。経験上，乱数の幅＝切り捨てる値の範囲がよいのではないかと私は考えますが，閾値はケースバイケースでいつも決定しています。

これで，なんで階調が表れるか？　と，なかなか納得できない人もいるでしょう。そういうときにはまず，グラデーションを想像します。グラデーションの2値化では途中，閾値でばっさり切られます。閾値が128だとしたら，120の箇所はばっさりとなくなります。もしもランダムディザがかかっていたらどうなるでしょうか？　120は8以上の乱数が出るだけで，切り上げられます。仮に0～127までの乱数をディザとして加算する場合，8以上であるケースが大半になるでしょう。100ならば28以上，50ならば78以上。つまりは閾値に近いほど，確率的に画素が残るケースが多くなり，より遠いほど(小さいほど)，画素が残るケースが少なくなります。要するに，確率の大小は，画素周辺の平均濃度になるわけです。

この例では閾値以上を考えていません。それには，閾値を切り捨てる最大値まで上げ，切り捨てた分だけランダムディザとして加算すれば同じ考えが適応できます。具体的にいうなら，0～255までの2値化なら，閾値を255にして，0～255を加算するといったところでしょうか。

ただ，注意しなくてはならないのは，X680x0のFLOAT?.xによって生成される乱数は，かなり片寄りがあることです。したがって，X680x0でランダムディザをかける場合，妙にパターン化されたノイズが乗るので，注意してください。

ランダムディザは一般にあまり使われていないようです。どちらかというとこの次で説明する，マトリクスディザのほうが多いようです。その大義名分は，画素を残すか残さないかを分ける値がランダムでは頼りないということらしいです。解説書などにもノイジーになるのであまり使われない，とあっさりと片付けられています。

私はどちらかというと，ランダムを求めるよりも，あらかじめ決まったマトリクスを足し算したほうが速いからじゃないだろうか？　と思っているのですが。

しかし，次に説明するマトリクスディザの場合，閾値から輝度変化の確率が同じ部分ではかならず同じパターンが現れるという法則性があります。この法則性のあるパターンが私には機械的に見えて，非常に気になるのですけどね。どちらかというと，印刷に応用するならばランダムディザのほうが私は好きかな。

2）　マトリクスディザ

マトリクスディザは加算するディザ数を行列式で表しています。多いのは，ベイヤー(Bayer)法といって，4×4の場合，

$$\begin{Bmatrix} 0 & 8 & 2 & 10 \\ 12 & 4 & 14 & 6 \\ 3 & 11 & 1 & 9 \\ 15 & 7 & 13 & 5 \end{Bmatrix}$$

という行列を利用します。見てわかるとおり，行列はたすき状に値が決まっています。256階調の画像を2値化する場合これを8×8(＝256)の行列に拡張します。

ランダムディザは，それぞれに画素が無関係にディザ数を加算しましたが，マトリクスの場合，画面に直接このマトリクスを埋め込みます。つまり，ベイヤー法ならば，(x,y)には0を，(x＋1,y)には8を，(x,y＋1)ならば12を，(x＋1,y＋1)には4を加算するのです。この場合，4×4ですから，(x＋4,y)の場合，ループして再び0を加算します。

そして，16階調のとき，輝度8の部分は，8以上が生き残ります。4×4の領域全体，輝度8であるとき，パターンはもうわかりましたよね？

$$\begin{Bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{Bmatrix}$$

このように見覚えのあるディザパターンが残るのです。サンプルを見ると，いたるところにこのようなパターンが見えますよね？

その他のディザパターンには，網点型，渦巻型などがあります。

網点型は，

$$\begin{Bmatrix} 11 & 4 & 6 & 9 \\ 12 & 0 & 2 & 14 \\ 7 & 8 & 10 & 5 \\ 3 & 15 & 13 & 1 \end{Bmatrix}$$

渦巻型は,

$$\begin{Bmatrix} 6 & 7 & 8 & 9 \\ 5 & 0 & 1 & 10 \\ 4 & 3 & 2 & 11 \\ 15 & 14 & 13 & 12 \end{Bmatrix}$$

のような配列に差し替えるだけで実現できます。

3)　濃度パターン法

濃度パターン法は，ディザとは多少違いますが，出力結果が割と近いので，これに含めて紹介することにします。たとえば，元画像の1画素を，2×2などのパターンに割り当てます。2×2の場合,

	○○ ○○	●○ ○○	●○ ○●	●● ○●	●● ●●
階調	0	1	2	3	4

このように，5階調が作れます。1画素が2×2になるので，元の画像よりも，面積が4倍になってしまう欠点があります。たとえば257階調ならば，1画素が8×8になるので，面積が256倍になってしまいます。

よく考えてみると2倍された画像に対して，2×2の行列をもったベイヤー法ディザをかけると，結果は同じになりますよね。現在では，元画像の解像度よりプリンタの解像度が格段に高くなっていますので，ある程度大きな絵を出すのであればこのような方法が効率よくなってきます。

4)　濃度パターン法＋ディザ法

濃度パターン法が面積において，現実的ではないのでこれとディザを組み合わせた方法が存在します。いままでのディザはあくまでも2値化を考えていましたが，濃度パターンの2×2を5階調として，ディザ行列を考えるのです。この方法でも結局ディザパターンが見えますが，現在のプリンタのように，解像度が妙に高くなってくると，有効かもしれません。濃度パターン＋マトリクスディザでは，法則性が似ているので，あまり意味がないかもしれませんが，濃度パターン＋ランダムなどではそれなりに面白い結果が出そうです。

ただし，濃度パターンはパターンの分だけサイズをとりますから，720dpiプリンタは，360dpi 5階調プリンタと理屈では等価となってしまいます。

また，濃度パターン法の場合，1ドットが完全に孤立しますから，原画像とプリンタの画素数に大きな隔たりがある場合，印字がくっきりとして，綺麗な印刷になります。SX-WINDOWでは，COPYキーを押して画面のハードコピーをした結果が，濃度パターン法です。

●誤差分散

原画像のあるピクセルの輝度をF(x,y)，その画素を単純に2値化した先の画素をG(x,y)とします。F(x,y)からG(x,y)変換したので，この誤差EはExy＝F(x,y)－G(x,y)で表されます。この誤差を，隣接する画素に振り分けるのが，誤差分散です。分散式は,

$$F'(x+1,y)=F(x+1,y)+Exy\times 3/8$$
$$F'(x+1,y+1)=F(x+1,y+1)+Exy\times 1/4$$
$$F'(x,y+1)=F(x,y+1)+Exy\times 3/8$$

で表します。

分散した画素は，積み重ねられていきます。つまり，このように1画素誤差分散したあと，次の画素F(x+1,y)に移るときには，すでに加算された誤差を踏まえて演算します。したがって，いちばん上の最初の画素は，分散率に影響されませんが，次の画素から左の画素の影響を受けた計算になります。

このようにアルゴリズム上，誤差分散はいちばん最初のラスタは影響を受けにくいので，素直に変換するといちばん上だけ様子が変わってしまいます。そのため，自分自身の誤差を自分に対して重みとして加えたりします。誤差の分散はCRTのスキャンのように左から右へ，上から下への落ちていく方式が主体です。

バリエーションとして，誤差の分散を行う場所の選択，誤差の分散比，そしてラスタの順序などを変えたりします。ラスタの順序を偶数ラインと奇数ラインで，右から左，左から右へと変える方法なども存在します。桑野式の場合，誤差の分散を左下の画素まで行い，それぞれの分散比がこれと少しずつ変わっているものです。具体的には,

$$F'(x+1,y)=F(x+1,y)+Exy\times 1/2$$
$$F'(x+1,y+1)=F(x+1,y+1)+Exy\times 1/8$$
$$F'(x,y+1)=F(x,y+1)+Exy\times 1/8+(Exy\ \%8)$$
$$F'(x-1,y+1)=F(x-1,y+1)+Exy\times 1/4$$

と変化させます。スキャン方式は左から右が基本ですが，原則というわけではないようです。

誤差分散の特色としては，隣接したドットに影響を与えるため，隣接したドットがつながるということです。また，ディザのようなタイルパターンはなく，人間の目から見てかなり自然なドットのばらまきをします。

分散率を変えると，ドットがつながり，ヒゲのようなものが出ます。解像度が低いとヒゲは太くなり気持ち悪く残ります。最近のレーザープリンタの場合，勝手にスムージングを行うので，誤差分散のヒゲがスムージングされ，ときおり，気持ち悪い分散をすることがあります。

SX-WINDOWにはFloyd-Steinberg誤差分散というものがありますが，これも分散率のノウハウのひとつなのでしょう。標準添付のものはバグっているようですが……。

白黒の印刷

カラー印刷とモノクロ印刷は，RGB 3色分変換するかしないかという点だけで，減色方法はさして変わりません。同じように，誤差分散し，ディザ変換をかけたりします。

ただし，プリンタドライバがカラーからモノクロにすると，ときおり納得できない変換をすることがあります。それを防ぐためには，画像を一度カラーからモノクロに変換しないとなりません。モノクロに変換するアルゴリズムはいくつかありますが，基本は色座標の変換です。たいていは，色平面と輝度で表す色座標系に変換して，そのうちの輝度だけを流用します。

1)　RGB平均値化によるモノクロ化

原理は，RGBのそれぞれの画素の平均を出力するものです。特徴としては，全体的に暗っぽい結果となります。濃度ヒストグラムは中域に片寄ります。この理由は，RGBの原色，すなわちR,G,Bの各色は，同じ強さで表現したとしても，人間にはBよりもR，RよりもGのほうが明るく感じるためでしょう。この方式は，このことを考慮しないので，画像は全体的にのっぺりとし，暗くなってしまいます。

2)　YIQのY（輝度）

1)の平均値による方法がのっぺりとして全体的に暗くなるなら，R,G,Bそれぞれに重みを与えてみるのがいいのではないだろうか？　という考えから，いちばん暗いBを基準にして，Rに2，Gに4の重みを与えてみました。

式は(2R＋4G＋1B)/7というもの。この重み，1,2,4は，いろいろ適当に試行錯誤をした挙句，2進数の1,2,4に落ち着きました。本当は8で割って高速化したかったのですが，出力結果を見ると，どうもうまい数値がありません。7でも計算量も少ないし，かなり自然な感じにモノクロ化します。

実はこれは，RGBから輝度Y，色差I，Q

で表せるYIQ座標系に変換し，輝度信号Yを求める計算とほぼ等価です。YIQの変換式は，$Y=0.299R+0.587G+0.144B$(ただしR,G,Bは正規化され，出力Yは0～1.03)で表されますが，この式のうちBの添字を1とすると，先の式にほぼ近くなるのです。

余談ではありますが，よくよく考えてみると，X1時代からの読者なら周知の0～7のカラーコードは，モノクロモニタから受け継がれているものです。なにも考えずに0～7のコードにあのカラーを配分したわけではないのだなあと，これを実験しつつ，ちょっと感心しました。

3) RGBのGreen

2)でGreenの係数が大きいので，Greenだけを利用する簡単なアルゴリズムを作っても，それなりにモノクロ画像が表現されるのではないでしょうか？　その結果，色の数が少ない塗り絵のような絵には効かないけれど，写真，フォトタッチ，パステルタッチで描かれたような色の多い絵には，思っていたよりも綺麗に出力することがわかりました。

割と人間の目はいい加減なものかもしれません。もちろん，情報を2つも単純に切り捨ててしまうため，ほかのモノクロ化アルゴリズムに比べると頼りないし，入力画像を選ぶのはしかたがないと思われます。しかし，計算はないに等しいので，リアルタイムでも使えます。

あとでわかったことですが，どうも，印刷結果の出来栄えから，SX-WINDOWのプリンタドライバは，遅いくせにこの方式を使っているような気がします。

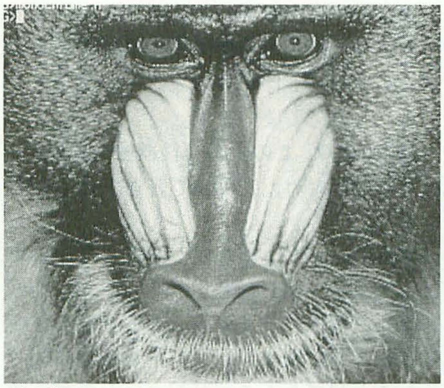
4) HSVのS（彩度）

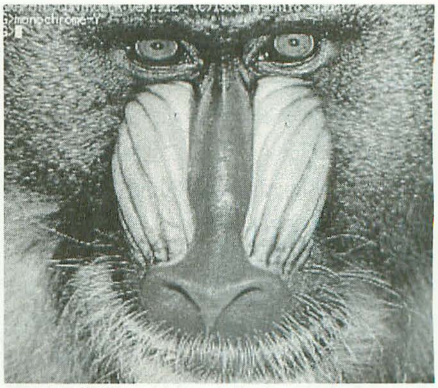
お馴染み，HSV座標に変換して，S（彩度または飽和度）だけを利用するアルゴリズムを利用してみました。この変換では彩度Sは，白レベルともいえ，明度Vは黒レベルと考えることもできます（厳密には違いますが)。普通は当然Vの値を使ってやるわけですが，「白っぽさ」に着目した場合の実験です。Sは0のとき真っ白で，正規化した場合は1のとき，白要素がありません。つまり印刷では黒ベタになります。

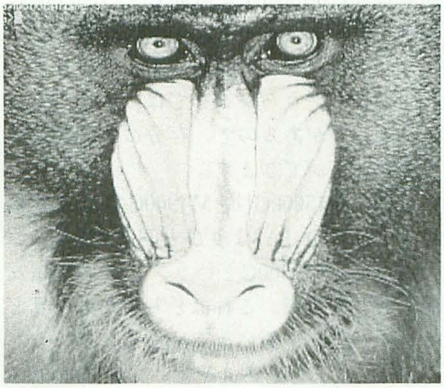
試しにやってはみたものの，やはりこの変換はあまりモノクロ化には向いていないようで，全体的に白っぽくなる傾向があります。

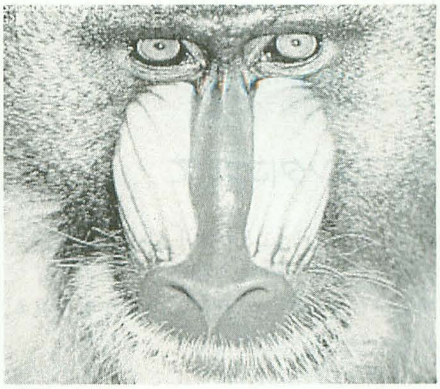
*　　*　　*

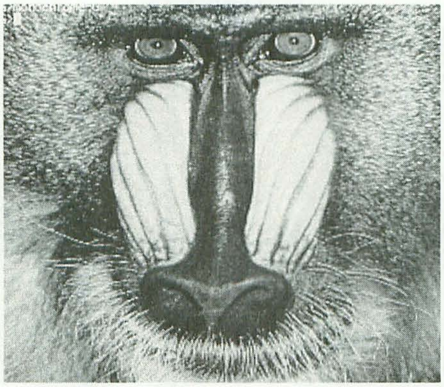
実は，La*b*表色系など，ほかにも座標系を試してみたですが，際だって綺麗だったものはありませんでした。

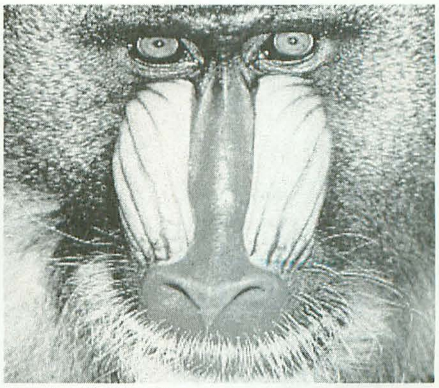

元画像


RGBの平均値


簡易型のY


Sを使ったもの


La＊B＊のL


きっちり計算したY

モノクロの場合，色は，黒寄りであるか，白寄りであるか？　で表せます。つまり，画像の中の固有物の判別は，黒か白かで判別するため，白と黒の輝度差がないと，印刷しても美しくなりません。

そのため，濃度ヒストグラムを利用します。これは，たとえば，輝度が0～255までなら，横軸を0～255にとり，画面全体の色をピックアップして，それぞれの濃度にどのぐらいの画素があるかを数えたものです。これを行うと，濃度分布がわかるので，もっとも暗い色，もっとも明るい色を，強制的に最大値と最小値に分け，輝度差を大きくできるのです。また，濃度ヒストグラムが谷になった部分は，もっとも閾値として向いている場所です。これを見つけ出し，ない場合は面積を2分するところを利用すると，綺麗な2値化をすることができます。

減色処理に関してですが，レーザープリンタの場合，誤差分散よりもディザのほうがよいと思います。理由は先に挙げたとおりです。

SXのプリンタドライバ

印刷環境が最低といわれてきた，SX-WINDOWの環境ですが，今月あたりに発売される，シャーペンワープロパック2の登場で，やっと，綺麗な印刷ができるようになりました。

サンプルは，シャーペンに張り付けた画像の印刷ですが，減色処理もかなり綺麗に

行っています。減色は誤差分散とディザです。実は，このパック2で，マッハジェット/エスパーマッハプリンタ用のドライバと，BJ用のドライバがシステムに追加されます。SXのドライバとして追加されるということは，Easydrawも720dpiとして印刷できますし，X-DTPだってかなり綺麗に印刷できます。

また，高速なディザ変換，誤差分散の原色リソースが追加されますから，レーザープリンタなどで印刷するときも，綺麗に印刷ができます。たぶん，MJユーザーの方は，やっとまともな印刷ができるようになった……と思うでしょう。

しかも，RGBの色補正，黒混合比などいろいろ調整があったりするからもう……。

プリンタについて

どうでもいいことかもしれませんが，エプソンのプリンタは，相当ラインナップが増えてしまって，どれがどのような存在かわからなくなってきてしまいました。

まず，基準はMJ700V2Cです。このプリンタは初代720dpiプリンタで，スーパーファイン専用紙のみ，720dpiで印刷できます。試しに普通紙へ印刷すると，インクがでろでろに染みてしまいます。

MJ800Cは，この正当な後継機種です。モノクロ印刷だけ，720dpiで普通紙に印刷できます。

MJ900Cはこれをさらに拡張した形です。よく見ると，MJ900Cはマッハジェットではなく，エスパーマッハと名前を変えていることがわかります。ドットをさらに絞り込むことができるモードがついたので，普通紙にも720dpi印字が可能です。700，800に比べると，スーパーファインに印刷したときも，印字はさらに美しくなっています。今回のワープロパック2では，エスパーマッハモードがあるので，存分にこの性能は生かすことができます。

そしてMJ5000Cは，MJ900CのA3ノビ版になるのです（A3よりひと回り大きい）。

別口で，MJ500Cというプリンタが超安価でありますが，これはどちらかというと，モノクロプリンタにカラーモードというものをつけた，むしろ，キヤノンのBJ400Jの対抗機種といえます。

近日，MJ3000Cなるものが出るそうですが，いま現在，情報はまだ入っておりません。噂では，A2対応というとてつもなくでかいプリンタになるとか……しまった，MJ5000Cを買ったのは早まったかなあとも思いましたが，エスパーマッハではなくマッハジェットであるとのこと。一度エスパーマッハをみてしまうと，マッハジェットはなあ……。

終わりに

なんだかわからないけど，死にそうに忙しくって，腕の筋がズキズキ痛いこの頃。おかげで，サンプルプログラムを作ろうかと思っていましたが，これも間にあいませんでした。すみません。

それでも，ワープロパック2があるから，とりあえず綺麗な印刷ができますよね。ワープロでグラフィックの印刷だけやるってのもなんか変ですけども。

リスト1

```
/*
Program     : gaussian.C
Version     : 0.01
Author      : Yasushi Taki

$Id$
$Log$

*/

//#define   __DEBUG_K_

#define     __DIRECT_FPU__

#include    <stdio.h>
#include    <stdlib.h>
#include    <sys/iocs.h>
#include    <sys/dos.h>
#include    <process.h>
#include    <math.h>



#define     GVRAM           0xc00000
#define     NEAR            4
#define     MAXCOLOR        256

typedef struct double_data{                     // double 型 data の構造体
    double ddata1;
    double ddata2;
    double ddata3;
} double_data;

typedef struct data{                            // int 型 data の構造体
    int    data1;
    int data2;
    int data3;
} data;


#define     TIMES   24

#define     putpoint(X,Y,R,G,B)     vram[(Y)][(X)]=((int)((B)>>3) << 1)+((int)((R)>>3) << 6)
+((int)((G)>>3) << 11);
#define     MAX(X,Y)        ((X)>(Y)?(X):(Y))
#define     MAX3(X,Y,Z)     ((Z)>(MAX(X,Y))?(Z):(MAX(X,Y)))
#define     MIN(X,Y)        ((X)<(Y)?(X):(Y))
#define     MIN3(X,Y,Z)     ((Z)<(MIN(X,Y))?(Z):(MIN(X,Y)))

#define     flab(A_A0)      (((A_A0)> 0.008856)? pow(A_A0,1./3) : 7.787*A_A0 + 16./116)

typedef     short   short512[512];              // VRAMを強引に配列にするための裏技

void        conv(void);
void        Gaussian(data ,int ,int ,double ,double );
data        torgb(int);                         // 15ビットカラーから、data型を返す。

void main( int argc,char **argv )
{

 conv();

}

void conv( )
{
    int                     stack;              // supervisor用stack
    data                    rgb;                // RGB int型
    register                int     x=0;
    int                             y=0;
    short512                *vram=GVRAM;
    data                    BVRAM[20][20];

    stack=_dos_super( 0 ); /* Super visor mode */
    y=0;
    while( y<=19){
        x=0;
        while(x<=19){
            rgb=torgb((int)vram[y][x]);
            BVRAM[y][x].data1=rgb.data1;
            BVRAM[y][x].data2=rgb.data2;
            BVRAM[y][x].data3=rgb.data3;
            x++;
        }
        y++;
    }

    _iocs_g_clr_on();

    y=0;
    while( y<=19){
        x=0;
        while(x<=19){
            Gaussian(BVRAM[y][x],(x+1)*TIMES,(y+1)*TIMES,0.5,TIMES);
            x++;
        }
        y++;
    }

    _dos_super(stack);
}

void        Gaussian(data rgb,int x_,int y_,double foot,double times)
/*
 rgbはdata型struct、フルカラーデータ。
 x_,y_は、画面上の中心座標点。この点を中心に、倍率*裾野率を利用して、
 ガウシャンパルスの裾野の幅を決めます。

  ガウシャンパルスを利用した拡大法はまだ研究の余地がある部分です。
デモを見ればわかると思いますが、どちらかというと、印刷する場合は、
CMY座標系に変換してからかけたほうがよいと思われます。
  アルゴリズムの特質上、ベタにモコモコの模様が出来てしまうのが問
題点です。メタボールのように隣接した山が重なるような理論をいま考
えているとこです。
  ただ、このデモでは20*20のマトリクスを、480*480へ、
24倍していますが、この計算には040turboのライトスルーで4分かか
ります（コピーバックだと2分37秒）。もしも、このアルゴリズムを、
画面全体にそのまま応用するとなると時間的にかなりつらくなるとおも
います。簡略近似式を作って高速化しないと、実際に使うのは、むずか
しいかもしれません。
*/
{
    double          t=0;
    data            rgb2;
    double          tau;
    data            temp;
    int                     xp,yp;
    short512        *vram=GVRAM;
    double          theta=0;
    double          sqx,sqy;

    tau=times*foot;                     // 広がる裾野長さを決定する。
    for(theta=0;theta<M_PI;theta+=0.03){
        t=-2*tau;                       // 2*2*tauは横幅
```

```
		while(t<=2*tau){			// timesは倍率
			sqx=cos(theta)*t;
			sqy=sin(theta)*t;
			if((abs((int)sqx)<=(int)(times/2)) && (abs((int)sqy)<=(int)(times/2))){
								// ワンショット領域を窓関数にする
				xp=(int)(x_-sqx);
				yp=(int)(y_-sqy);
//					rgb2=torgb((int)vram[yp][xp]);	// 輝度呼びだし(実験で利用した)
				rgb2.data1=0;
				rgb2.data2=0;
				rgb2.data3=0;
				temp.data1=(int)(rgb.data1*exp(-(t*t)/(2*tau*tau)));// GaussianPaluseの計算
				temp.data2=(int)(rgb.data2*exp(-(t*t)/(2*tau*tau)));
				temp.data3=(int)(rgb.data3*exp(-(t*t)/(2*tau*tau)));
//					printf("画面上の元:%d,%d,%d¥n",rgb2.data1,rgb2.data2,rgb2.data3);
//					printf("退避VRAM上:%d,%d,%d¥n",rgb.data1 ,rgb.data2 ,rgb.data3);
//					printf("   計算後:%d,%d,%d¥n",temp.data1,temp.data2,temp.data3);
				temp.data1+=rgb2.data1;
				temp.data2+=rgb2.data2;
				temp.data3+=rgb2.data3;
				if(temp.data1>255) temp.data1=255;
				if(temp.data2>255) temp.data2=255;
				if(temp.data3>255) temp.data3=255;
//					printf(" 足し算後:%d,%d,%d¥n¥n",temp.data1,temp.data2,temp.data3);
				putpoint(xp,yp,temp.data1,temp.data2,temp.data3);
			}
			t++;
		}
	}
}

data		torgb(int dat)
{
	data	rgb;
	rgb.data1 = (((dat & 0x07c0)>> 6)<<3); // RED
	rgb.data2 = (((dat & 0xf800)>>11)<<3); // GREEN
	rgb.data3 = (((dat & 0x003e)>> 1)<<3); // BLUE
	return(rgb);
}
```

リスト2

```
#include	<stdio.h>
#include	<stdlib.h>
#include	<sys/dos.h>
#include	<sys/iocs.h>
#include	<sys/xglob.h>
#include	<string.h>
#include	<process.h>

/*
	$Id: to16.c,v 1.2 1994/12/07 20:58:27 Kohju Exp $
	$Log: to16.c,v $
 * Revision 1.2  1994/12/07  20:58:27  Kohju
 * ・分割ソースにした
 * ・マトリクスをパレットごとに4つにふやした
 * ・ゼロ補間、直線補間による単純閾値法を追加した。
 *
*/

#define	GVRAM				0xc00000

typedef	short	short512[512];
typedef	short	short1024[1024];

	void	pallet( int );
	void	matrix( int ,const unsigned short exvram[512][512]);
	void	単純閾値法(int,const unsigned short exvram[512][512]);
	void	ディザ(int,const unsigned short exvram[512][512]);

/******** program start ********/

void main(int argc,char **argv)
{
	int	FourClr=0;					// 4階調にする色はどれか?
								// 0=すべて2階調, 1=青, 2=赤, 4=緑
	int	Zelo=0;						// ゼロ補間を行う。
	short512	*vram=(short512 *)GVRAM;		// VRAM(superでのみアクセスすること)
	int			ix,iy,c,stk;			// 変数群
static unsigned short	exvram[512][512];

	printf("t16.x 6万色512×512→16色1024×1024化プログラム  by 灘¥n");

	while((c = getopt(argc,argv,"8gGrRbBhHIiTtZzDd")) != EOF) {
		switch (c) {
		case '8':
			printf("8色拡散を行います。¥n");
			FourClr=0;
			break;
		case 'b':
		case 'B':
			printf("青のみ4階調にし、16色拡散を行います。¥n");
			FourClr=1;
			break;
		case 'r':
		case 'R':
			printf("赤のみ4階調にし、16色拡散を行います。¥n");
			FourClr=2;
			break;
		case 'g':
		case 'G':
			printf("緑のみ4階調にし、16色拡散を行います。¥n");
			FourClr=4;
			break;
		case 'i':
		case 'I':
			printf("輝度を含めて、16色拡散を行います。¥n");
			FourClr=8;
			break;
		case 't':
		case 'T':
			printf("単純閾値法にて変換します。¥n");
			FourClr='T';
			break;
		case 'z':
		case 'Z':
			printf("ゼロ補間を行います。¥n");
			Zelo=1;
			break;
		case 'd':
		case 'D':
			printf("ディザ変換で8色に減色します。¥n");
			FourClr='D';
			break;
		case 'h':
		case 'H':
		case '?':
			printf(		"オプション説明:(nは整数です)¥n"
				" -B 濃度パターン法:5階調減色で8色に変換する(デフォルト)¥n"
				" -R 濃度パターン法:赤のみ9階調にし、5階調減色で16色に変換する。¥n"
				" -G 濃度パターン法:緑のみ9階調にし、5階調減色で16色に変換する。¥n"
				" -B 濃度パターン法:青のみ9階調にし、5階調減色で16色に変換する。¥n"
				" -I 濃度パターン法:輝度を含めて16色に変換をする。¥n"
				"  -D ディザ変換で8色に減色します。¥n"
				"  -T 単純閾値法で8色に減色します。¥n"
				"  -Z マトリクス減色以外の方法でゼロ補間を行う。デフォルトは直線補間¥n"
				"  -h このヘルプをみます¥n");
			exit(-1);
			break;
		}
	}
	argc -= optind;
	argv = &argv[optind];

	stk=_dos_super(0);
// スーパバイザモードに移行

	for(iy=0;iy<=511;iy++){
		for(ix=0;ix<=511;ix++){				// VRAMをテンポラリにバックアップ
			exvram[iy][ix]=(unsigned short)vram[iy][ix];
		}
	}
	*(unsigned char  *)0xe80028=(unsigned char )0x0004;	// VICON R20
	*(unsigned short *)0xe82400=(unsigned short)0x0004;	// VIPS  R00 16色モードへ
											//
ハード的に変換……

	pallet(FourClr);
	_dos_super(stk);
// ユーザモードに移行

	switch(FourClr){
		case 0:
		case 1:
		case 2:
		case 4:
		case 8:
			matrix(FourClr,exvram);
			break;
		case 'T':
			単純閾値法(Zelo,exvram);
			break;
		case 'D':
			ディザ(Zelo,exvram);
			break;
	}
}
```

リスト3

```
void		pallet(int FourClr)
{
//		共通パレット

	*(unsigned short *)0xe82000=(unsigned short)0b00000_00000_00000_0;
	*(unsigned short *)0xe82002=(unsigned short)0b00000_00000_11111_0;
	*(unsigned short *)0xe82004=(unsigned short)0b00000_11111_00000_0;
	*(unsigned short *)0xe82006=(unsigned short)0b00000_11111_11111_0;
	*(unsigned short *)0xe82008=(unsigned short)0b11111_00000_00000_0;
	*(unsigned short *)0xe8200a=(unsigned short)0b11111_00000_11111_0;
	*(unsigned short *)0xe8200c=(unsigned short)0b11111_11111_00000_0;
	*(unsigned short *)0xe8200e=(unsigned short)0b11111_11111_11111_0;

// RGBパレット
	switch(FourClr){
		case 0 :
			break;
		case 1 :
			*(unsigned short *)0xe82010=(unsigned short)0b00000_00000_00000_0;//中間青
			*(unsigned short *)0xe82012=(unsigned short)0b00000_00000_01111_0;
			*(unsigned short *)0xe82016=(unsigned short)0b00000_11111_01111_0;
			*(unsigned short *)0xe82014=(unsigned short)0b00000_11111_00000_0;
			*(unsigned short *)0xe82018=(unsigned short)0b11111_00000_00000_0;
			*(unsigned short *)0xe8201a=(unsigned short)0b11111_00000_01111_0;
			*(unsigned short *)0xe8201c=(unsigned short)0b11111_11111_00000_0;
			*(unsigned short *)0xe8201e=(unsigned short)0b11111_11111_01111_0;
		break;
		case 2 :
			*(unsigned short *)0xe82010=(unsigned short)0b00000_00000_00000_0;//中間赤
			*(unsigned short *)0xe82012=(unsigned short)0b00000_00000_11111_0;
			*(unsigned short *)0xe82016=(unsigned short)0b00000_01111_11111_0;
			*(unsigned short *)0xe82014=(unsigned short)0b00000_01111_00000_0;
			*(unsigned short *)0xe82018=(unsigned short)0b11111_00000_00000_0;
			*(unsigned short *)0xe8201a=(unsigned short)0b11111_00000_11111_0;
			*(unsigned short *)0xe8201c=(unsigned short)0b11111_01111_00000_0;
			*(unsigned short *)0xe8201e=(unsigned short)0b11111_01111_11111_0;
		break;
		case 4 :
		case 'T' :
		case 'D' :
			*(unsigned short *)0xe82010=(unsigned short)0b00000_00000_00000_0;//中間緑
			*(unsigned short *)0xe82012=(unsigned short)0b00000_00000_11111_0;
			*(unsigned short *)0xe82014=(unsigned short)0b00000_11111_00000_0;
			*(unsigned short *)0xe82016=(unsigned short)0b00000_11111_11111_0;
			*(unsigned short *)0xe82018=(unsigned short)0b01111_00000_00000_0;
			*(unsigned short *)0xe8201a=(unsigned short)0b01111_00000_11111_0;
			*(unsigned short *)0xe8201c=(unsigned short)0b01111_11111_00000_0;
			*(unsigned short *)0xe8201e=(unsigned short)0b01111_11111_11111_0;
		break;
		case 8 :
```

```
            *(unsigned short *)0xe82010=(unsigned short)0b01111_01111_01111_0;//輝度
            *(unsigned short *)0xe82012=(unsigned short)0b01111_01111_11111_0;
            *(unsigned short *)0xe82016=(unsigned short)0b01111_11111_11111_0;
            *(unsigned short *)0xe82014=(unsigned short)0b01111_11111_01111_0;
            *(unsigned short *)0xe82018=(unsigned short)0b11111_01111_01111_0;
            *(unsigned short *)0xe8201a=(unsigned short)0b11111_01111_11111_0;
            *(unsigned short *)0xe8201c=(unsigned short)0b11111_11111_01111_0;
            *(unsigned short *)0xe8201e=(unsigned short)0b11111_11111_11111_0;
        break;
    }
}
```

リスト4

```
#include    <stdio.h>
#include    <sys/xglob.h>

#define     GVRAM                       0xc00000

typedef     short   short512[512];
typedef     short   short1024[1024];

void        matrix(int FourClr,const unsigned short exvram[512][512])
{
unsigned short      dat=0;                              // 読み込みデータ
unsigned short      dat2[4]={0,0,0,0};
    int                     ix,iy,r,g,b,i,k,stk;    // 変数群
    short1024       *vram16=(short1024 *)GVRAM;     // VRAM(superでのみアクセスすること)

    stk=_dos_super(0);                              // スーパバイザモードに移行
    printf(         "2×2濃度パターン法減色中(supervisor modeで作業します)・・・¥n");
    for(iy=0;iy<=511;iy++){
        for(ix=0;ix<=511;ix++){
            dat=exvram[iy][ix];                     // VRAMからデータを抽出。
            dat=dat >> 1;                   // 15ビットカラーから別々に5ビットづつ分けた
            b=(int)(dat & 0x1f);
            r=(int)((dat >> 5) & 0x1f);
            g=(int)((dat >> 10) & 0x1f);
//                  _print("¥n¥rg=%d,r=%d,b=%d",b,r,g);
            for(k=0;k<=3;k++) dat2[k]=0;            // dat2の初期化
            switch(FourClr){
                case 0 :
                    if(b>5)         dat2[0]+=1;
                    if(b>11)dat2[3]+=1;
                    if(b>18)dat2[2]+=1;
                    if(b>24)dat2[1]+=1;
                    if(r>5)         dat2[1]+=2;
                    if(r>11)dat2[0]+=2;
                    if(r>18)dat2[3]+=2;
                    if(r>24)dat2[2]+=2;
                    if(g>5) dat2[3]+=4;
                    if(g>11)dat2[2]+=4;
                    if(g>18)dat2[1]+=4;
                    if(g>24)dat2[0]+=4;
                    break;                          // ひとつの色に付き4階調になる。
                                                    // 輝度を半分落とす時は8を足す。
                case 1 :
                    if(b> 2)dat2[0]+=9;
                    if(b> 5)dat2[3]+=9;
                    if(b> 9)dat2[2]+=9;
                    if(b>11)dat2[1]+=9;
                    if(b>14)dat2[0]-=8;
                    if(b>18)dat2[3]-=8;
                    if(b>21)dat2[2]-=8;
                    if(b>24)dat2[1]-=8;
                    if(r>5)         dat2[1]+=2;
                    if(r>11)dat2[0]+=2;
                    if(r>18)dat2[3]+=2;
                    if(r>24)dat2[2]+=2;
                    if(g>5)         dat2[3]+=4;
                    if(g>11)dat2[2]+=4;
                    if(g>18)dat2[1]+=4;
                    if(g>24)dat2[0]+=4;
                    break;
                case 2 :
                    if(b>5)         dat2[0]+= 1;
                    if(b>11)dat2[3]+= 1;
                    if(b>18)dat2[2]+= 1;
                    if(b>24)dat2[1]+= 1;
                    if(r> 2)dat2[1]+=10;
                    if(r> 5)dat2[0]+=10;
                    if(r> 9)dat2[3]+=10;
                    if(r>11)dat2[2]+=10;
                    if(r>14)dat2[1]-= 8;
                    if(r>18)dat2[0]-= 8;
                    if(r>21)dat2[3]-= 8;
                    if(r>24)dat2[2]-= 8;
                    if(g>5) dat2[3]+= 4;
                    if(g>11)dat2[2]+= 4;
                    if(g>18)dat2[1]+= 4;
                    if(g>24)dat2[0]+= 4;
                    break;
                case 4 :
                    if(b>5)         dat2[0]+=1;
                    if(b>11)dat2[3]+=1;
                    if(b>18)dat2[2]+=1;
                    if(b>24)dat2[1]+=1;
                    if(r>5)         dat2[1]+=2;
                    if(r>11)dat2[0]+=2;
                    if(r>18)dat2[3]+=2;
                    if(r>24)dat2[2]+=2;
                    if(g> 2)dat2[3]+=12;
                    if(g> 5)dat2[2]+=12;
                    if(g> 9)dat2[1]+=12;
                    if(g>11)dat2[0]+=12;
                    if(g>14)dat2[3]-= 8;
                    if(g>18)dat2[2]-= 8;
                    if(g>21)dat2[1]-= 8;
                    if(g>24)dat2[0]-= 8;
                    break;
                case 8 :
                    i=((( r < g )? r :g)< b )? (( r < g )? r :g):b;
                    if(b>5)         dat2[0]+=1;
                    if(b>11)dat2[3]+=1;
                    if(b>18)dat2[2]+=1;
                    if(b>24)dat2[1]+=1;
                    if(r>5)         dat2[1]+=2;
                    if(r>11)dat2[0]+=2;
                    if(r>18)dat2[3]+=2;
                    if(r>24)dat2[2]+=2;
                    if(g>5) dat2[3]+=4;
                    if(g>11)dat2[2]+=4;
                    if(g>18)dat2[1]+=4;
                    if(g>24)dat2[0]+=4;
                    if(i>05)dat2[3]+=8;
                    if(i>11)dat2[2]+=8;
                    if(i>18)dat2[1]+=8;
                    if(i>24)dat2[0]+=8;
                    break;                          // ひとつの色に付き4階調になる。
            }
            vram16[iy*2  ][ix*2  ]=dat2[0];
            vram16[iy*2  ][ix*2+1]=dat2[1];
            vram16[iy*2+1][ix*2  ]=dat2[2];
            vram16[iy*2+1][ix*2+1]=dat2[3];
        }
    }
    _dos_super(stk);                                // ユーザモードに移行
}
```

リスト5

```
* 単純閾値法の減色プログラム
*
* 引数)
*       int     Zero;                   ¥¥ ゼロ補間を行うか? 1ならばゼロ補間
*
   0ならば直線補間
*const unsigned short exvram[512][512]  ¥¥ 裏VRAMのアドレス

* int   単純閾値法( int Zero, const unsigned short exvram[512][512]);

        .include        doscall.mac

        .xdef   _単純閾値法

        .offset 4

Zero    ds.l    1       * (0)
exvram  ds.l    1       * (1)

        .text

savesize        equ     4*9                     * 9はレジスタのサイズ

_単純閾値法:
        movem.l d3-d7/a3-a6,-(sp)

*       全てのアドレスは使用可能

        movem.l Zero+savesize(a7),d0/a6
        move.l  #$C00000,a5

*               d0      ゼロ補間をするかどうか?
*               a5      VRAMのマップアドレス
*               a6      裏VRAMのマップアドレス(512KB)

        tst.l   d0
*       bne             _Zero                   * ゼロ補間を行う

        move.l  a6,d7
        add.l   #$80000,d7

loop:
        move.w  (a6)+,d0                        *ドットをVRAMから読み出して次のドットへ
        move.w  d0,d1
        andi.w  #%00000_00000_11111_0,d1        * d1はB
        lsr.w   #$01,d1
        move.w  d0,d2
        andi.w  #%00000_11111_00000_0,d2        * d2はR
        lsr.w   #$06,d2
        andi.w  #%11111_00000_00000_0,d0        * d0はG
        lsr.w   #$06,d0
        lsr.w   #$05,d0

        clr.w   d3
*       clr.w   d4
*       clr.w   d5
*       clr.w   d6

        cmpi.w  #$0f,d1
        blt             @f                      * Bが15以上なら以下を実行
        ori.w   #$01,d3                                 * その点
*       ori.w   #$01,d4                                 * その下の点
*       ori.w   #$01,d5                                 * その隣の点
*       ori.w   #$01,d6                                 * その隣の下の点
@@:
        cmpi.w  #$0f,d2
        blt             @f                      * Rが15以上なら以下を実行
        ori.w   #$02,d3                                 * その点
*       ori.w   #$02,d4                                 * その下の点
*       ori.w   #$02,d5                                 * その隣の点
*       ori.w   #$02,d6                                 * その隣の下の点
@@:
        cmpi.w  #$0f,d0
        blt             @f                      * Gが15以上なら以下を実行
```

▶最近GUNSN'ROSESとMETALLICAにハマってます。空耳アワー(タモリ倶楽部)の見過ぎでしょうか?
村上　浩二(20)京都府

```
         ori.w   #$04,d3                                  * その点
*        ori.w   #$04,d4                                  * その下の点
*        ori.w   #$04,d5                                  * その隣の点
*        ori.w   #$04,d6                                  * その隣の下の点

@@:
         clr.l   -(sp)
         DOS             __SUPER                  * スーパバヴァイザモードに移行
         addq.l  #$4,sp
         move.l  d0,a4

         move.w  d3,(a5)
         move.w  d3,$800(a5)
         move.w  d3,2(a5)
         move.w  d3,$802(a5)                      * 実際に書き込む

         move.l  a4,-(sp)
         DOS             __SUPER                  * ユーザーモードに戻る
```

```
         addq.l  #$4,sp

         addq.l  #$4,a5                           * 次のドットへ、

         move.l  a5,d3
         andi.l  #$7ff,d3
         tst.l   d3                               * X軸分のビットが0なら
         bne             @f
         add.l   #$800,a5                         * 0ドット目なら1ライン分下げる
@@:
         cmp.l   d7,a6
         blt             loop                     * 0だったら落ちる。
         bra             _End
_Zero:
_End:
         movem.l (sp)+,d3-d7/a3-a6
         rts
```

リスト6

```
* ディザの減色プログラム
*
*  引き数)
*         int     Zero;                    ¥¥ ゼロ補間を行うか？1ならばゼロ補間
*
  0ならば直線補間
*const unsigned short exvram[512][512]  ¥¥ 裏VRAMのアドレス

* int   ディザ( int Zero, const unsigned short exvram[512][512]);

Thured  equ     24

        .include        doscall.mac

        .xdef   _ディザ

        .offset 4

Zero    ds.l    1       * (0)
exvram  ds.l    1       * (1)

        .text

savesize        equ     4*9                     * 9はレジスタのサイズ

_ディザ:
        movem.l d3-d7/a3-a6,-(sp)

*       全てのアドレスは使用可能

        movem.l Zero+savesize(a7),d0/a6
        move.l  #$C00000,a5

*               d0      ゼロ補間をするかどうか？
*               a5      VRAMのマップアドレス
*               a6      裏VRAMのマップアドレス(512KB)

        tst.l   d0
*       bne             _Zero                   * ゼロ補間を行う

        move.l  a6,d7
        add.l   #$80000,d7

        clr.l   -(sp)
        DOS             __SUPER                 * スーパバヴァイザモードに移行
        addq.l  #$4,sp
        move.l  d0,a4

loop:
        move.w  (a6)+,d0                        * ドットをVRAMから読み出して次のドットへ
        move.w  d0,d1
        andi.w  #%00000_00000_11111_0,d1        * d1はB
        lsr.w   #$01,d1
        move.w  d0,d2
        andi.w  #%00000_11111_00000_0,d2        * d2はR
        lsr.w   #$06,d2
        andi.w  #%11111_00000_00000_0,d0        * d0はG
        lsr.w   #$06,d0
        lsr.w   #$05,d0

        clr.w   d3
        clr.w   d4
        clr.w   d5
        clr.w   d6

        cmpi.w  #Thured-0,d1                    * 0 2×2のディザ行列
        blt             @f                              * Bが18以上なら以下を実行
        ori.w   #01,d3                                  * その点
@@:     cmpi.w  #Thured-2,d1                    * 2
        blt             @f                              * Bが16以上なら以下を実行
        ori.w   #01,d6                                  * その隣の下の点
@@:     cmpi.w  #Thured-6,d1                    * 6
        blt             @f                              * Bが12以上なら以下を実行
        ori.w   #01,d5                                  * その隣の点
@@:     cmpi.w  #Thured-8,d1                    * 8
        blt             @f                              * Bが10以上なら以下を実行
        ori.w   #$01,d4                                 * その下の点

@@:     cmpi.w  #Thured-0,d2                    * 0 2×2のディザ行列
        blt             @f                              * Rが18以上なら以下を実行
        ori.w   #$02,d3                                 * その点
@@:     cmpi.w  #Thured-2,d2                    * 2
        blt             @f                              * Rが16以上なら以下を実行
        ori.w   #$02,d6                                 * その隣の下の点
@@:     cmpi.w  #Thured-6,d2                    * 6
        blt             @f                              * Rが12以上なら以下を実行
        ori.w   #$02,d5                                 * その隣の点
@@:     cmpi.w  #Thured-8,d2                    * 8
        blt             @f                              * Rが10以上なら以下を実行
        ori.w   #$02,d4                                 * その下の点

@@:     cmpi.w  #Thured-0,d0                            * 0 2×2のディザ行列
        blt             @f                              * Gが18以上なら以下を実行
        ori.w   #$04,d3                                 * その点
@@:     cmpi.w  #Thured-2,d0                    * 2
        blt             @f                              * Gが16以上なら以下を実行
        ori.w   #$04,d6                                 * その隣の下の点
@@:     cmpi.w  #Thured-6,d0                    * 6
        blt             @f                              * Gが12以上なら以下を実行
        ori.w   #$04,d5                         * その隣の点
@@:     cmpi.w  #Thured-8,d0                    * 8
        blt             @f                              * Gが10以上なら以下を実行
        ori.w   #$04,d4                                 * その下の点

@@:

        move.w  d3,(a5)
        move.w  d4,$802(a5)                             * 実際に書き込む
        move.w  d5,2(a5)
        move.w  d6,$800(a5)

        addq.l  #$4,a5                                  * 次のドットへ、

        move.l  a5,d3
        andi.l  #$7ff,d3
        tst.l   d3                                      * X軸分のビットが0なら
        bne             @f
        add.l   #$800,a5                                * 0ドット目なら1ライン分下げる。
@@:
        cmp.l   d7,a6
        blt             loop                            * 0だったら落ちる。
        bra             _End
_Zero:
_End:
        move.l  a4,-(sp)
        DOS             __SUPER                         * ユーザーモードに戻る
        addq.l  #$4,sp
        movem.l (sp)+,d3-d7/a3-a6
        rts
```

リスト7

```
/*
Program     : monocrome.C
Version     : 0.01
Author      : Yasushi Taki

$Id: monochrome.c,v 1.1 1995/08/02 14:39:16 Kohju Exp $
$Log: monochrome.c,v $
 * Revision 1.1  1995/08/02  14:39:16  Kohju
 * Initial revision
 *
 * Revision 1.2  1995/08/01  08:24:21  Kohju
 * ・data型を利用するように変形
 * ・Lab形式を埋め込む。
 *

*/

//#define  __DEBUG_K_

#define    __DIRECT_FPU__

#include   <stdio.h>
#include   <stdlib.h>
#include   <sys/iocs.h>
#include   <sys/dos.h>
#include   <process.h>
#include   <math.h>

#include   "glm.h"                             // GLMローダヘッダ

#define    GVRAM          0xc00000
#define    NEAR           4
#define    MAXCOLOR       256

typedef struct double_data{                    // double 型 data の構造体
    double  ddata1;
    double  ddata2;
    double  ddata3;
} double_data;

typedef struct data{                           // int 型 data の構造体
    int     data1;
    int data2;
    int data3;
} data;


#define            putpoint(X,Y,R,G,B)     vram[(Y)][(X)]=((int)((B)>>3) << 1)+((int)((R)>>
3) << 6)+((int)((G)>>3) << 11);
#define            MAX(X,Y)        ((X)>(Y)?(X):(Y))
#define            MAX3(X,Y,Z)     ((Z)>(MAX(X,Y))?(Z):(MAX(X,Y)))
#define            MIN(X,Y)        ((X)<(Y)?(X):(Y))
#define            MIN3(X,Y,Z)     ((Z)<(MIN(X,Y))?(Z):(MIN(X,Y)))
```

▶ X68000を買うべく通販の広告を見て電話をしたら，どこも売り切れでした。誰かいつでも売っているところを教えてください。　　松本　雅裕(21)岩手県

```
#define             flab(A_A0)  (((A_A0)> 0.008856)? pow(A_A0,1./3) : 7.787*A_A0 + 16./116)

typedef     short   short512[512];          // VRAMを強引に配列にするための裏技

data                torgb(int);             // 15ビットカラーから、data型を返す。
data                rgb_hsv(data);          // RGBからHSVに変換する
double_data         rgb_xyz(data );
//double            flab (double );
double_data         xyz_lab (double_data );
double_data         xyz_luv (double_data );


void main( int argc,char **argv )
{

    int option='A';                     // オプション
    int     inc;                        // インクリメント用変数
    int     fflag=0;                    // ファイル名が指定されたか? 1で指定
    char    temp[1024];                 // テンポラリ
    char    temp2[1024];                // テンポラリ
    char    ext[24];
    char    l_ext[24];                  // 拡張子
    int     crtmode=-1;                 // オプションcが設定されると、画面が最後に戻される。
                                                        //
    int     xsize=511,ysize=511;

    /*      オプション判別                      */
    for(inc=1;inc<argc;inc++) {
            if(argv[inc][0]=='-'){  /* オプションモード */
                    switch(argv[inc][1]){
                            case 'A':                                       // 平均値を求める場合
                                            option='A';
                                            break;
                            case 'Y':                                       // 輝度Y
                                            option='Y';
                                            break;
                            case 'S':                                       // 彩度S
                                            option='S';
                                            break;
                            case 'L':                                       // 輝度L
                                            option='L';
                                            break;
                            case 'G':                                       // GREENだけを利用する
                                            option='G';
                                            break;
                            case 'x':
                                            xsize=atoi(&argv[inc][2]);
                                            break;
                            case 'y':
                                            ysize=atoi(&argv[inc][2]);
                                            break;
                            case 'c':       // _iocs_crtmode(-1);を入れると、crtmodeに現在の値が
返る
                                        // 画面はそのまま
                                            crtmode=_iocs_crtmod(-1);
                                            break;
                            case 'h':
// ヘルプだよん
                            case '?':
                            default :
                                            help();
                                            exit(-1);
                                            break;
                    }
            }
            else {  // ファイルの検索
                    printf("画像ファイル %s¥n",argv[ inc ]);
                    getlastext( ext,l_ext,argv[inc]);
                    if(strcmpi(ext,"glm")==0){
                    }
                    else spawnlp(P_WAIT,"apic.r","apic.r",argv[inc],NULL);
                    conv(option,xsize,ysize);
                    sprintf(temp ,"-s0,0,%d,%d",xsize,ysize);
                    sprintf(temp2,"%s%c.%s",l_ext,option,ext);
                    if(strcmpi(ext,"glm")==0)       {
                    }
                    else spawnlp(P_WAIT,"apic.r","apic.r",temp,temp2,NULL);
                    fflag=1;
            }
    }
    if(fflag==0) conv(option,xsize,ysize);
    _iocs_crtmod(crtmode);
}

void conv( int opt ,int xsize,int ysize)
{
    int                     stack;          // supervisor用stack
    register        int     mono;
    data            rgb;                    // RGB int型
    register        int     x=0;
    int                     y=0;
    short512        *vram=GVRAM;


    stack=_dos_super( 0 );  /* Super visor mode */
    y=0;
    while( y<=ysize){
            x=0;
            while(x<=xsize){
                    rgb=torgb((int)vram[y][x]);
                    if(opt=='S') mono=MAX3(rgb.data1,rgb.data2,rgb.data3);
                                            //せっかくrgbtohsv作ったのになあ……
                    if(opt=='A') mono=(rgb.data1+rgb.data2+rgb.data3)/3;
                    if(opt=='G') mono=rgb.data2;
                    if(opt=='Y') mono=(2*rgb.data1+4*rgb.data2+rgb.data3)/7;
                    if(opt=='L') {
                            mono = (int)((116* flab(
                                    1.0000/255.0/100*(double)rgb.data1
                                  + 4.5907/255.0/100*(double)rgb.data2
                                  + 0.0601/255.0/100*(double)rgb.data3
                                  ) - 16)*255/28.5);
                    }
                    putpoint(x,y,mono,mono,mono);
                    x++;
            }
            y+=1;
    }
    _dos_super(stack);
}

void        help( void )
{
    printf( "Explain : 画像をモノクロ化させるためのツール¥n"
                    "¥n"
                    "       -A      RGBの平均値を利用する。¥n"
                    "       -Y      YIQの輝度Y¥n"
                    "       -S      HSVの彩度S¥n"
                    "       -L      Labの輝度L¥n¥n"
                    "       -G      Greenだけで変換する¥n"
                    );
}

data        torgb(int dat)
{
    data    rgb;
    rgb.data1 = (((dat & 0x07c0)>> 6)<<3);  // RED
    rgb.data2 = (((dat & 0xf800)>>11)<<3);  // GREEN
    rgb.data3 = (((dat & 0x003e)>> 1)<<3);  // BLUE
    return(rgb);
}


data        rgb_hsv(data rgb)
/*
    RGBからA.H.Munsell表色系HSVに変換する
    引数)
            rgb.data1       r
            rgb.data2       g
            rgb.data3       b
    戻り値)
            hsv.data1       h
            hsv.data2       s
            hsv.data3       v

    H:色相(hue)
      0~ 31 : 赤~黄色
     32~ 63 : 黄色~緑
     64~ 95 : 緑~シアン
     96~127 : シアン~青
    128~159 : 青~マゼンタ
    160~191 : マゼンタ~赤
    192以上  : ないはず……

    S:彩度(saturation)      白レベル        最大値はRGBの最大値MAXCOLORによる。
    V:明度()                黒レベル        最大値はRGBの最大値MAXCOLORによる。
*/

{
    int             chk=0;          // チェック用変数
    data    hsv;            // HSV

    hsv.data3=MAX3(rgb.data1,rgb.data2,rgb.data3);
    chk=hsv.data3-MIN3(rgb.data1,rgb.data2,rgb.data3);
    if ( chk!=0 ) {
            hsv.data2=chk*MAXCOLOR/hsv.data3;
            if(hsv.data3==rgb.data1  ) hsv.data1=(rgb.data2-rgb.data3)*MAXCOLOR/chk;
            if(hsv.data3==rgb.data2  ) hsv.data1=(rgb.data3-rgb.data1)*MAXCOLOR/chk+64;
            if(hsv.data3==rgb.data3  ) hsv.data1=(rgb.data1-rgb.data2)*MAXCOLOR/chk+128;
            if(hsv.data1< 0  ) hsv.data1+=192;
            if(hsv.data1<192 ) hsv.data1-=192;
    }
    else {
            hsv.data1=0;
            hsv.data2=0;
    }
}

double_data rgb_xyz(data rgb)
{
    double_data xyz;

    xyz.ddata1 = 2.7689/255.0*(double)rgb.data1 + 1.7517/255.0*(double)rgb.data2 + 1.1302/25
5.0*(double)rgb.data3;
    xyz.ddata2 = 1.0000/255.0*(double)rgb.data1 + 4.5907/255.0*(double)rgb.data2 + 0.0601/25
5.0*(double)rgb.data3;
    xyz.ddata3 =                (double)rgb.data1 + 0.0565/255.0*(double)rgb.data2 + 5.5943/25
5.0*(double)rgb.data3;
    return(xyz);
}

double_data xyz_lab (double_data xyz)
{
    double_data     lab;

    lab.ddata1 = (int)(116* flab(xyz.ddata2/100) - 16);
    lab.ddata2 = (int)(500*(flab(xyz.ddata1/100) - flab(xyz.ddata2/100)));
    lab.ddata3 = (int)(200*(flab(xyz.ddata2/100) - flab(xyz.ddata3/100)));
    return(lab);
}

double_data xyz_luv (double_data xyz)
{
    double_data     luv;
    double          luv_l;

    luv_l = 116*flab(xyz.ddata2/100)-16;
    luv.ddata1 = (int)luv_l;
    luv.ddata2 = (int)(13*luv_l*(4*xyz.ddata1 / (xyz.ddata1 + 15*xyz.ddata2 + 3*xyz.ddata3)
- 4./19));
                            /* (4/19)=[(4*x/(x+15*y+3*z))にx=y=z=100を代入したもの] */
    luv.ddata3 = (int)(13*luv_l*(9*xyz.ddata2 / (xyz.ddata1 + 15*xyz.ddata2 + 3*xyz.ddata3)
- 9./19));
                            /* (9/19)=[(9*y/(x+15*y+3*z))にx=y=z=100を代入したもの] */
    return(luv);
}

int         getlastext( char *ext, char *limit_ext,const char *name)
/*
機能:
    name が指すファイル名にから、
    最後に付加された拡張子を取り出し、ext に格納して返します。
    また、拡張子を除外したファイル名を、limit_ext に格納して返します。
戻り値:
    最後のピリオドのポインタを返します。
    拡張子がまったくなかった場合は -1 を返します。
*/

{
    int     len=0,si=0,di=0,end=0,result=-1,i=0;
    si=len=(strlen(name)-1);
    do{
            if(name[si]=='.') {
                    result=si;
                    si++;
                    while(name[si]!=0){
                            ext[di]=name[si];
                            di++;
                            si++;
                    }
                    ext[di]=0;
                    for(i=0;i<result;i++) limit_ext[i]=name[i];
                    limit_ext[i]=0;
                    end=1;
            }
            if(name[si]=='/' ) {result=-1;end=1;}
            if(name[si]=='¥¥') {result=-1;end=1;}
            if(name[si]==':' ) {result=-1;end=1;}
            si--;
            if(si<=0) {result=-1;end=1;}
    } while(end!=1);
    return(result);
}
```

MJ700V2C/5000C用印刷プログラム

HCESCV2I.X

Ikeda Mutsumi 池田 睦

**1994年10月号のHCESCV2Xをもとに作成されたハードコピーツールだ
変換部がアセンブラで記述されA3対応，高速，省メモリを実現した
16色，256色，フルカラーにまで対応している**

瀧康史氏のHCESCV2Xを独自に機能強化しました。一部暫定機能もありますが，HCESCV2Xの機能強化版のような位置付けになっています。主要部はアセンブラで書き直しました。A4いっぱいのサイズでも1時間ちょっとで印刷します(X68000XVI)。また，1ラインごとに処理するのでメモリをほとんど必要としなくなりました。

なお，色変換には対応していません。

使用方法

コマンドラインから，

A> HCESCV2I ［オプション］［ファイル名］

で起動。オプションには，以下のものがあります。

- -Sn サイズの指定（n：0～4）
- -Mn 色分解モード
  - n＝0 3色分解
  - n＝3 4色分解
  - n＝4 白黒分解
- -N 画面を初期化しない
- -Pn 領域指定
  - n＝0 画面全域
  - n＝1 左上1/4
  - n＝2 右上1/4
  - n＝3 左下1/4
  - n＝4 右下1/4
- -An 用紙指定
  - n＝3 A3に印刷
  - n＝4 A4に印刷
- -Fn 紙送り
  - n＝0 印刷後紙送りしない
  - n＝1 印刷後紙送り実行
- -H 水平方向
- -V 垂直方向
- -DYn 黄色濃度調整(標準＝5，0が濃い)
- -DMn マゼンタ濃度調整
- -DCn シアン濃度調整
- -DBn 黒濃度調整

HCESCV2Xにあった-C，-Bオプションはサポートしていません。

-Nオプションでは16色，256色でのページ0の印刷も可能です。

ファイル名にはGL3，IMG，RGB(点順次)，GLX，GLM，PICが指定可能です。PICファイルを指定すると内部でAPICを呼び出します。

元絵の解像度は直線補間によって解像度上げられています。RGBそれぞれ11ビットとして扱われます。

4色分解は瀧氏の方式を再現できなかったので独自のものを使っています。

減色には桑野式拡散を使用しています。ただし，下方向への分配は，

n×1/8＋(n mod 8)

ではなく，単に，

n×1/8

で表現しています。輝度それぞれを11ビットで扱っている関係上，(n mod 8)はそれほど影響してこないでしょう。

暫定となっている元画像の整数倍でしか印刷できないとか，元画像のアスペクト比を無視して紙の大きさにあわせるとか，問題があれば各自で改善してみてください。

リスト1 main.c

```
#include	<stdio.h>
#include	<string.h>

#include	"hcescv2i.h"

struct	cmstat	{						/*コマンドオプション情報*/
		short		hv_flag;			/*印刷方向 0=標準 1=水平垂直反転*/
		short		b_flag;				/*色分解法 0-4*/
		short		ff_flag;			/*改ページ 0=しない 1=する*/
		struct ymcbstat sikii;			/*しきい値 0=濃い 5=標準 9=薄い*/
		short		size;				/*印刷用紙サイズ 3=A3 4=A4*/
		short		page;				/*印刷対称範囲 0=全体 1-4=1/4領域*/
		short		opt_s;				/*印刷サイズ 0-3*/
		short		q_flag;				/*印刷条件確認 0=しない 1=する*/
	};
void			getpar(short *par,char *,short,short);	/*数値の受取*/
short			hcopy(struct cmstat *,char *);			/*印刷*/

struct	rgbstat *GETRGB_GL3(int,struct hcstat *);		/*GL3ファイルの印刷*/
struct	rgbstat *GETRGB_IMG(int,struct hcstat *);		/*IMGファイルの印刷*/
struct	rgbstat *GETRGB_RGB(int,struct hcstat *);		/*RGBファイルの印刷*/
struct	rgbstat *GETRGB_GLX(int,struct hcstat *);		/*GLXファイルの印刷*/
struct	rgbstat *GETRGB_GLM(int,struct hcstat *);		/*GLMファイルの印刷*/

struct ymcbstat	*RGBTOYMC(struct rgbstat *);			/*三色分解*/
struct ymcbstat	*RGBTOYMCB_TY(struct rgbstat *);		/*瀧康史氏による四色分解（未完成）*/
struct ymcbstat	*RGBTOBLACK(struct rgbstat *);			/*黒単色*/

struct ymcbstat	SI[5]={									/*各色分解のしきい値（黒マシ黄の順）*/
				{0x7fff,0x1400,0x1400,0x1400},	/*三色分解の場合のしきい値*/
				{0x7fff,0x7fff,0x7fff,0x7fff},
				{0x7fff,0x7fff,0x7fff,0x7fff},
				{0x0e00,0x1200,0x1200,0x1200},	/*四色分解の場合のしきい値*/
				{0x1000,0x7fff,0x7fff,0x7fff},	/*黒単色のしきい値*/
			};

void
main(argc,argv,envp)
int				argc;
char			**argv;
char			**envp;
	{
	struct	cmstat	hc[1];
	char			*par,com;
	short			i;

	B_SUPER(0);				/*VRAMの印刷を考えてスーパーバイザーモードにする*/
	hc->hv_flag=0;			/*画像データの水平方向=プリンタの水平方向*/
	hc->b_flag=3;			/*色分解=四色分解*/
	hc->ff_flag=1;			/*改ページ=する*/
	hc->size=4;				/*用紙サイズ=A4*/
	hc->page=0;				/*印刷範囲=全体*/
	hc->opt_s=3;			/*印刷サイズ=用紙限界*/
	hc->q_flag=0;			/*印刷条件確認=する*/

	hc->sikii.b=5;			/*      黒のしきい値=標準*/
	hc->sikii.m=5;			/*マゼンダのしきい値=標準*/
	hc->sikii.c=5;			/*  シアンのしきい値=標準*/
	hc->sikii.y=5;			/*    黄色のしきい値=標準*/

	for (i=1;i<argc;++i) {				/*コマンドオプションの解析*/
		par=argv[i];
		if (*par!='-') {				/*ファイル名指定であれば*/
			hcopy(hc,argv[i]);			/*指定されたファイルを印刷し*/
			continue;					/*次のオプションを読む*/
		}
		++par;
		com=(*par++);
		if (com>='A' && com<='Z') com=com-'A'+'a';
		switch (com) {
			case 'n' :					/*Nオプションなら*/
				hcopy(hc,NULL);			/*ファイル指定なしで印刷*/
				break;
			case 'd' :					/*しきい値の設定*/
				com=(*par++);
				switch (com) {
					case 'B' :
					case 'b' : getpar(&hc->sikii.b,par,0,9); break;
					case 'M' :
					case 'm' : getpar(&hc->sikii.m,par,0,9); break;
					case 'C' :
					case 'c' : getpar(&hc->sikii.c,par,0,9); break;
					case 'Y' :
					case 'y' : getpar(&hc->sikii.y,par,0,9); break;
```

```c
                }
                break;
            case 'h' : if (*par=='¥0') hc->hv_flag=0; break;            /*水平=水平*/
            case 'v' : if (*par=='¥0') hc->hv_flag=1; break;            /*水平=垂直*/
            case 'm' : getpar(&hc->b_flag ,par,0,4); break;             /*色分解モード設定*/
            case 'f' : getpar(&hc->ff_flag,par,0,1); break;             /*改ページフラグ設定*/
            case 'q' : getpar(&hc->q_flag ,par,0,1); break;             /*印刷条件確認設定*/
            case 'a' : getpar(&hc->size   ,par,3,4); break;             /*用紙サイズ設定*/
            case 'p' : getpar(&hc->page   ,par,0,4); break;             /*印刷範囲設定*/
            case 's' : getpar(&hc->opt_s  ,par,0,3); break;             /*印刷サイズ設定*/
            case '?' :
                printf("HCESCV2i.x : High Colors' Hard copy program for MJ-700V2C/5000C¥n");
                printf("                        Copyright (c)'95 by Mutumi Ikeda¥n");
                printf("HCESCV2i [option・・・see bellow] filename [option・・・see bellow] filename [option・・・see below]¥n");
                printf("Explain : 拡散方式は桑野式です。¥n");
                printf("Optional switches:¥n");
                printf(" -N  : ファイルをロードせず、画面もクリアしません。(コプロセスやSXコンソール内で有効)¥n");
                printf(" -Mn : 変換モードの指定。0:3色変換 3:4色変換 4:黒単色¥n");
                printf(" -Sn : 印刷サイズ。      0:ノーマル 1:横印刷   2:正方形  3:縦印刷¥n");
                printf(" -An : 用紙サイズ         3:A3      4:A4¥n");
                printf(" -Fn : 印刷後の紙送り     0:しない   1:する¥n");
                printf(" -Qn : 印刷前の確認       0:しない   1:する¥n");
                printf(" -Pn : 印刷対象領域       0:全体     1:左上1/4  2:右上1/4  3:左下1/4  4:右下1/4¥n");
                printf("File formats:¥n");
                printf(" .PIC : 内部でAPICが呼ばれ画像サイズは512x512と判断します¥n");
                printf(" .MAG : 内部でmag が呼ばれ画像サイズは640x400と判断します¥n");
                printf(" .GL3 : ¥n");
                printf(" .IMG : ¥n");
                printf(" .RGB : 同一ディレクトリに.IPRファイルが必要です¥n");
                printf(" .Q0  : 同一ディレクトリに.IPRファイルが必要です¥n");
                printf(" .GLX : ¥n");
                printf(" .GLM : ¥n");
                break;
        }
    }
}

void
getpar(par,data,min,max)
short               *par;
char                *data;
short               min,max;
    {
    if (*data=='¥0') return;
    if (*data=='=') ++data;
    if (data[1]=='¥0' && *data>='0'+min && *data<='0'+max) {
        *par=(*data)-'0';
    }
}

short
hcopy(hc,filename)
struct      cmstat *hc;
char                *filename;
    {
    FILE            *ptr;
    struct  hcstat  hs;
    short           xs,ys,size,ret;
    int             q,i,j;
    char            com[256];
    struct ymcbstat ymcb;
    unsigned short  *us;
    unsigned char   header[256]={"    "};
    char            file[256],iprfile[256];

    if (filename==NULL) {
        us=(unsigned short *)0xe82400;
        if (*us & 0x0004) hs.xs=hs.ys=1024;
        else              hs.xs=hs.ys= 512;
        switch (*us & 0x0003) {
            case 0 : hs.color=0x0010; break;
            case 1 : hs.color=0x0100; break;
            case 3 : hs.color=0x8000; break;
        }
        hs.data=(void *)0xc00000;
        HCSET_DF(GETRGB_VRAM);
    } else {
        strcpy(file,filename);
        for (i=0;i<strlen(file);++i) {
            if (file[i]>='a' && file[i]<='z') file[i]=file[i]+'A'-'a';
        }
        ptr=fopen(file,"rb");
        if (ptr==NULL) {
            printf("File can't open.¥n");
            return (-1);
        } else {
            fread(header,1,6,ptr);
            if (strlen(file)>=4 && memcmp(file+strlen(file)-4,".PIC",4)==0) {
                hs.xs=hs.ys=512;
                hs.color=0x8000;
                hs.data=(void *)0xc00000;
                HCSET_DF(GETRGB_VRAM);
                fclose(ptr);
                sprintf(com,"apic %s",file);
                system(com);
            } else if (strlen(file)>=4 && memcmp(file+strlen(file)-4,".MAG",4)==0) {
                hs.xs=640;
                hs.ys=400;
                hs.color=0x10;
                hs.data=(void *)0xc00000;
                HCSET_DF(GETRGB_VRAM);
                fclose(ptr);
                sprintf(com,"mag %s",file);
                system(com);
            } else if (strlen(file)>=4 && memcmp(file+strlen(file)-4,".GL3",4)==0) {
                hs.xs=512;
                hs.ys=(filelength(fileno(ptr))/1024);
                hs.color=0x8000;
                hs.data=(void *)file;
                HCSET_DF(GETRGB_GL3);
                fclose(ptr);
            } else if (strlen(file)>=4 && memcmp(file+strlen(file)-4,".IMG",4)==0) {
                hs.xs=(header[3]<<8) | header[2];
                hs.ys=(header[5]<<8) | header[4];
                hs.color=0x1000000;
                hs.data=(void *)file;
                HCSET_DF(GETRGB_IMG);
                fclose(ptr);
            } else if (strlen(file)>=4 && (memcmp(file+strlen(file)-4,".RGB",4)==0 || memcmp(file+strlen(file)-3,".Q0",3)==0)) {
                fclose(ptr);
                strcpy(iprfile,file);
                strcpy(iprfile+strlen(iprfile)-3,"IPR");
                ptr=fopen(iprfile,"rb");
                if (ptr==NULL) return (-1);
                hs.xs=hs.ys=0;
                fread(header,1,filelength(fileno(ptr)),ptr);
                for (i=0,j=0;i<filelength(fileno(ptr));++i) {
                    if (header[i]>='0' && header[i]<='9') {
                        if (j==0) hs.xs=hs.xs*10+header[i]-'0';
                        else      hs.ys=hs.ys*10+header[i]-'0';
                    } else {
                        ++j;
                        if (j==2) break;
                    }
                }
                fclose(ptr);
                hs.color=0x1000000;
                hs.data=(void *)file;
                HCSET_DF(GETRGB_RGB);
            } else if (strlen(file)>=4 && memcmp(file+strlen(file)-4,".GLX",4)==0) {
                hs.xs=(header[1]<<8)|header[0];
                hs.ys=(header[3]<<8)|header[2];
                hs.color=0x8000;
                hs.data=(void *)file;
                HCSET_DF(GETRGB_GLX);
                fclose(ptr);
            } else if (memcmp(header,"GR65",4)==0) {
                fseek(ptr,0x000c,0);
                fread(&hs.xs,2,1,ptr);
                fread(&hs.ys,2,1,ptr);
                hs.color=0x8000;
                hs.data=(void *)file;
                HCSET_DF(GETRGB_GLM);
                fclose(ptr);
            } else return (-1);
        }
    }

    if (hc->page>=0) {
        switch (hc->page) {
            case 0 : hs.x1=     0,hs.y1=     0,hs.x2=hs.xs  -1,hs.y2=hs.ys  -1; break;
            case 1 : hs.x1=     0,hs.y1=     0,hs.x2=hs.xs/2-1,hs.y2=hs.ys/2-1; break;
            case 2 : hs.x1=hs.xs/2,hs.y1=     0,hs.x2=hs.xs  -1,hs.y2=hs.ys/2-1; break;
            case 3 : hs.x1=     0,hs.y1=hs.ys/2,hs.x2=hs.xs/2-1,hs.y2=hs.ys  -1; break;
            case 4 : hs.x1=hs.xs/2,hs.y1=hs.ys/2,hs.x2=hs.xs  -1,hs.y2=hs.ys  -1; break;
        }
    }
    if (hs.x1>hs.x2) hs.x1^=hs.x2,hs.x2^=hs.x1,hs.x1^=hs.x2;
    if (hs.y1>hs.y2) hs.y1^=hs.y2,hs.y2^=hs.y1,hs.y1^=hs.y2;
    hs.wxs=hs.x2-hs.x1+1;
    hs.wys=hs.y2-hs.y1+1;
    if (hs.wxs<2 || hs.wys<2) return (-1);

    hs.ph=A4H;
    hs.pv=A4V;
    if (hc->size & 1) hs.ph^=hs.pv,hs.pv^=hs.ph,hs.ph^=hs.pv,hs.pv<<=1;
    size=(hc->size+1)>>1;
    if (size<2) hs.ph<<=(2-size),hs.pv<<=(2-size);
    if (size>2) hs.ph>>=(size-2),hs.pv<<=(size-2);
    if (hc->opt_s==1) hs.pv>>=1;

    hs.hv_flag=hc->hv_flag;
    hs.ff_flag=hc->ff_flag;
    switch (hc->opt_s) {
        case 0 : hs.hv_flag= 0,hs.h_zoom= 3,hs.v_zoom= 2; break;
        case 1 : hs.hv_flag= 0,hs.h_zoom=-1,hs.v_zoom=-1; break;
        case 2 :
        case 3 : hs.hv_flag=-1,hs.h_zoom=-1,hs.v_zoom=-1; break;
    }
    if (hs.hv_flag<0 && hs.h_zoom<0) {
        if (hs.color>16 && hc->opt_s!=2) {
            hs.hv_flag=(hs.wxs*3<hs.wys*2 ? 0 : 1);
        } else {
            hs.hv_flag=(hs.wxs  <hs.wys   ? 0 : 1);
        }
    } else if (hs.hv_flag<0 && hs.h_zoom>0) {
        if (hs.wxs*hs.h_zoom>hs.ph) hs.hv_flag=1;
    }
    if (hs.h_zoom<0) {
        if (hs.hv_flag) {
            hs.h_zoom=hs.ph/hs.wys;
            hs.v_zoom=hs.pv/hs.wxs;
        } else {
            hs.h_zoom=hs.ph/hs.wxs;
            hs.v_zoom=hs.pv/hs.wys;
        }
    }
    if (hc->opt_s==2) {
        if (hs.h_zoom>hs.v_zoom) hs.h_zoom=hs.v_zoom;
        else                     hs.v_zoom=hs.h_zoom;
    }
    if (hs.h_zoom<1 || hs.v_zoom<1) return (-1);

    ymcb.b=SI[hc->b_flag].b*(hc->sikii.b+5)/10;
    ymcb.m=SI[hc->b_flag].m*(hc->sikii.m+5)/10;
    ymcb.c=SI[hc->b_flag].c*(hc->sikii.c+5)/10;
    ymcb.y=SI[hc->b_flag].y*(hc->sikii.y+5)/10;
    HCSET_SIKII(&ymcb);
    switch (hc->b_flag) {
        case 0 : HCSET_CF(RGBTOYMC  ); break;
        case 1 :
        case 2 :
        case 3 : HCSET_CF(RGBTOYMCB ); break;
        case 4 : HCSET_CF(RGBTOBLACK); break;
    }

    if (hs.hv_flag) {
        hs.x =hs.x2;
        hs.y =hs.y1;
        hs.hx= 0;
        hs.hy= 1;
        hs.vx=-1;
        hs.vy=-hs.wys;
        hs.h_size=hs.wys;
        hs.v_size=hs.wxs;
    } else {
        hs.x =hs.x1;
        hs.y =hs.y1;
        hs.hx= 1;
        hs.hy= 0;
        hs.vx=-hs.wxs;
        hs.vy= 1;
        hs.h_size=hs.wxs;
        hs.v_size=hs.wys;
    }

    if (hc->q_flag) {
        printf("印刷対象 ");
        if (filename==NULL) printf("VRAM¥n");
        else printf("ファイル(%s)¥n",filename);
        printf("画像サイズ %d,%d¥n",hs.xs,hs.ys);
        printf("色数 %d¥n",hs.color);
        printf("範囲 (%d,%d)-(%d,%d)¥n",hs.x1,hs.y1,hs.x2,hs.y2);
        printf("拡大率 ");
        if (hs.hv_flag) printf("%d,%d¥n",hs.v_zoom,hs.h_zoom);
        else            printf("%d,%d¥n",hs.h_zoom,hs.v_zoom);
        if (hs.hv_flag) printf("元画像の水平方向=プリンタの垂直方向¥n");
        else            printf("元画像の水平方向=プリンタの水平方向¥n");
        printf("用紙サイズ ");
        if (hc->size==3)      printf("A3 ");
        else if (hc->size==4) printf("A4 ");
        printf("(%d,%d)¥n",hs.ph,hs.pv);
```

▶ 3月から和歌山で本屋の社員になりました。Oh!Xが入っていなかったので2冊ずつ入れ始めましたが，やはり買うのは私だけ。少し悲しい。　工藤　圭一(29)和歌山県

```
        switch (hc->opt_s) {
            case 0 : printf("小さいサイズ¥n"); break;
            case 1 : printf("用紙の半分¥n"); break;
            case 2 : printf("正方形¥n"); break;
            case 3 : printf("大きいサイズ¥n"); break;
        }
        switch (hc->b_flag) {
            case 0 : printf("三色分解¥n"); break;
            case 1 :
            case 2 :
            case 3 : printf("四色分解¥n"); break;
            case 4 : printf("黒単色  ¥n"); break;
        }
        printf("    黒のしきい値 %d¥n",hc->sikii.b);
        printf("マゼンダのしきい値 %d¥n",hc->sikii.m);
        printf("  シアンのしきい値 %d¥n",hc->sikii.c);
        printf("    黄色のしきい値 %d¥n",hc->sikii.y);
        printf("作業用メモリ %dバイト¥n",hs.h_size*hs.h_zoom*4*(hs.v_zoom+3)+((hs.h_size*hs.v_zoom+15)&0xfffe)/8*4);
        printf("印刷後の紙送り");
        if (hs.ff_flag) printf("する  ¥n");
        else            printf("しない¥n");
        q=' ';
        while (q!='y' && q!='Y') {
            printf("Ok?(y/n)");
            q=getche();
            printf("¥n");
            if (q=='n' || q=='N') return (0);
        }
    }
    ret=HCESCV2I(&hs);

    switch (ret) {
        case -1 : printf("プリンタの準備ができてません。¥n"); break;
        case -2 : printf("メモリ不足です。             ¥n"); break;
        case -3 : printf("処理能力を越えてます。       ¥n"); break;
    }
    return (ret);
}

/*RGB抽出の関数*/
/*RGB抽出の関数はHCESCV2I()から次の場合呼ばれる*/
/*印刷前、印刷中、改行後、印刷後*/
/*それぞれの場合に対応した処理を行う*/
/*印刷前・必要であれば画像データを返すのに必要な前処理を行う*/
/*印刷中・印刷に必要な画像データをRGBの値で返す*/
/*改行後・必要であれば改行に対する処理を行う*/
/*印刷後・必要であれば後処理を行う*/
/*関数の形式を守れば以下の関数以外に他のRGB抽出の関数を追加することができる*/
/*追加した関数は HCSET_DF(関数名); を行った後、HCESCV2I()を呼ぶことによって使用可能*/
/* struct rgbstat *関数名(command,x,y)                          */
/*          int     command;                                    */
/*                  0=印刷中*/
/*                  1=印刷前*/
/*                  2=改行後*/
/*                  3=印刷後*/
/*          int     x;                                          */
/*          int     y;                                          */
/*                                                              */

struct     rgbstat RGB;

struct     rgbstat *
GETRGB_GL3(command,st)
int                command;
struct     hcstat  *st;
   {
   FILE              *ptr;
   unsigned short  col,*data;

   switch (command) {
       case 0 :
           fseek((FILE *)st->data,(st->y<<10)+(st->x<<1),0);
           fread(&col,2,1,(FILE *)st->data);
           RGB.green=(col & 0xf800)>>5;
           RGB.red  =(col & 0x07c0)    ;
           RGB.blue =(col & 0x003e)<<5;
           RGB.green|=(RGB.green>>5)|(RGB.green>>10);
           RGB.red  |=(RGB.red  >>5)|(RGB.red  >>10);
           RGB.blue |=(RGB.blue >>5)|(RGB.blue >>10);
           break;
       case 1 :
           ptr=fopen(st->data,"rb");
           if (GETRGB_RAM2B(command,st)==0) {
               fread(st->data,2,(size_t)(st->xs*st->ys),ptr);
               fclose(ptr);
               HCSET_DF(GETRGB_RAM2B);
           } else {
               st->data=(void *)ptr;
           }
           break;
       case 3 :
           fclose((FILE *)st->data);
           break;
   }
   return(&RGB);
}

struct     rgbstat *
GETRGB_IMG(command,st)
int                command;
struct     hcstat  *st;
   {
   FILE              *ptr;
   unsigned char   col[3],*data;

   switch (command) {
       case 0 :
           fseek((FILE *)st->data,14+(st->xs*st->y+st->x)*3,0);
           fread(col,1,3,(FILE *)st->data);
           RGB.red  =(col[0])<<3;
           RGB.green=(col[1])<<3;
           RGB.blue =(col[2])<<3;
           RGB.green|=(RGB.green>>8);
           RGB.red  |=(RGB.red  >>8);
           RGB.blue |=(RGB.blue >>8);
           break;
       case 1 :
           ptr=fopen(st->data,"rb");
           if (GETRGB_RAM3B(command,st)==0) {
               fseek(ptr,14,0);
               fread(st->data,3,(size_t)(st->xs*st->ys),ptr);
               fclose(ptr);
               HCSET_DF(GETRGB_RAM3B);
           } else {
               st->data=(void *)ptr;
           }
           break;
       case 3 :
           fclose((FILE *)st->data);
           break;
   }
   return(&RGB);
}

struct     rgbstat *
GETRGB_RGB(command,st)
int                command;
struct     hcstat  *st;
   {
   FILE              *ptr;
   unsigned char   col[3],*data;

   switch (command) {
       case 0 :
           fseek((FILE *)st->data,(st->xs*st->y+st->x)*3,0);
           fread(col,1,3,(FILE *)st->data);
           RGB.red  =(col[0])<<3;
           RGB.green=(col[1])<<3;
           RGB.blue =(col[2])<<3;
           RGB.green|=(RGB.green>>8);
           RGB.red  |=(RGB.red  >>8);
           RGB.blue |=(RGB.blue >>8);
           break;
       case 1 :
           ptr=fopen(st->data,"rb");
           GETRGB_RAM3B(command,st);
           if (st->data!=NULL) {
               fread(st->data,3,(size_t)(st->xs*st->ys),ptr);
               fclose(ptr);
               HCSET_DF(GETRGB_RAM3B);
           } else {
               st->data=(void *)ptr;
           }
           break;
       case 3 :
           fclose((FILE *)st->data);
           break;
   }
   return(&RGB);
}

struct     rgbstat *
GETRGB_GLX(command,st)
int                command;
struct     hcstat  *st;
   {
   FILE              *ptr;
   unsigned short  col,*data;

   switch (command) {
       case 0 :
           fseek((FILE *)st->data,4+(st->xs*st->y+st->x)*2,0);
           fread(&col,2,1,(FILE *)st->data);
           RGB.green=(col & 0xf800)>>5;
           RGB.red  =(col & 0x07c0)    ;
           RGB.blue =(col & 0x003e)<<5;
           RGB.green|=(RGB.green>>5)|(RGB.green>>10);
           RGB.red  |=(RGB.red  >>5)|(RGB.red  >>10);
           RGB.blue |=(RGB.blue >>5)|(RGB.blue >>10);
           break;
       case 1 :
           ptr=(void *)fopen(st->data,"rb");
           GETRGB_RAM2B(command,st);
           if (st->data!=NULL) {
               fseek(ptr,4,0);
               fread(st->data,2,(size_t)(st->xs*st->ys),ptr);
               fclose(ptr);
               HCSET_DF(GETRGB_RAM2B);
           } else {
               st->data=(void *)ptr;
           }
           break;
       case 3 :
           fclose((FILE *)st->data);
           break;
   }
   return(&RGB);
}

struct     rgbstat *
GETRGB_GLM(command,st)
int                command;
struct     hcstat  *st;
   {
   FILE              *ptr;
   unsigned short  col,*data;

   switch (command) {
       case 0 :
           fseek((FILE *)st->data,16+(st->xs*st->y+st->x)*2,0);
           fread(&col,2,1,(FILE *)st->data);
           RGB.green=(col & 0xf800)>>5;
           RGB.red  =(col & 0x07c0)    ;
           RGB.blue =(col & 0x003e)<<5;
           RGB.green|=(RGB.green>>5)|(RGB.green>>10);
           RGB.red  |=(RGB.red  >>5)|(RGB.red  >>10);
           RGB.blue |=(RGB.blue >>5)|(RGB.blue >>10);
           break;
       case 1 :
           ptr=(void *)fopen(st->data,"rb");
           GETRGB_RAM2B(command,st);
           if (st->data!=NULL) {
               fread(st->data,2,(size_t)(st->xs*st->ys),ptr);
               fclose(ptr);
               HCSET_DF(GETRGB_RAM2B);
           } else {
               st->data=(void *)ptr;
           }
           break;
       case 3 :
           fclose((FILE *)st->data);
           break;
   }
   return(&RGB);
}


/*色分解の関数*/
/*色分解の関数はHCESCV2I()からRGB抽出後に呼ばれる*/
/*RBG各輝度から黒・マゼンダ・シアン・黄色の濃さを返す*/
/*関数の形式を守れば以下の関数以外に他の色分解の関数を追加することができる*/
/*追加した関数は HCSET_CF(関数名); を行った後、HCESCV2I()を呼ぶことによって使用可能*/
/* struct ymcbstat *関数名(green,red,blue)                      */
/*          int     green;  緑の輝度                 */
/*          int     red;    赤の輝度                 */
/*          int     blue;   青の輝度                 */
/*                                                              */

struct ymcbstat    YMCB;                      /*色分解の情報を返すための領域*/

struct ymcbstat *
RGBTOYMC(rgb)                                 /*三色分解*/
struct     rgbstat *rgb;
   {
   YMCB.b=0;                        /*黒は不用*/
```

▶内容がだんだん技術書みたいになってきた。　菊永　孝義(25)兵庫県

```
    YMCB.m=rgb->green^RGB_MAX;        /*マゼンダは緑の反転*/
    YMCB.c=rgb->red  ^RGB_MAX;        /* シアンは赤の反転*/
    YMCB.y=rgb->blue ^RGB_MAX;        /*   黄色は青の反転*/
    return(&YMCB);
}

struct ymcbstat *                     /*黒単色*/
RGBTOBLACK(rgb)
struct     rgbstat *rgb;
    {
    YMCB.b=((rgb->green^RGB_MAX)+(rgb->red^RGB_MAX)+(rgb->blue^RGB_MAX))/3;
    YMCB.m=0;
    YMCB.c=0;
    YMCB.y=0;
    return(&YMCB);
}

struct ymcbstat *
RGBTOYMCB_TY(rgb)  /*瀧康史氏による四色分解(後処理の関係上使用できない)*/
struct     rgbstat *rgb;
    {
/* 黒は、CMYのすべてが大きな時、(黒めの色になってきた時)多く拡散される。   */
/* CMYお互いが近いほど、(灰色に近いほど)多く拡散される。                    */

    int     min,max;                  /* 最小値と最大値。          */
    int     z;                        /* カウンタ                  */
    int     dif;                      /* 最大と最小の差            */

    YMCB.m=rgb->green^RGB_MAX;
    YMCB.c=rgb->red  ^RGB_MAX;
    YMCB.y=rgb->blue ^RGB_MAX;

    if (YMCB.m==YMCB.c && YMCB.m==YMCB.y) {
        YMCB.b=YMCB.m;
    } else {
        min=(YMCB.m<YMCB.c ? YMCB.m : YMCB.c);      /* 最大値と最小値をとる。*/
        min=(min   <YMCB.y ? min    : YMCB.y);
        max=(YMCB.m>YMCB.c ? YMCB.m : YMCB.c);
        max=(max   >YMCB.y ? max    : YMCB.y);
        dif=max-min;                                /* 最大値と最小値の差を求める。 */
        YMCB.b=(min*(min/RGB_MAX)+min*(RGB_MAX-dif/RGB_MAX));
/*             ^^^^^^^^^^^^^^^^                     CMYすべてが大きい時      */
/*                                  ^^^^^^^^^^^^^^^^^^^^  CMYがお互いに近い  */
        if (YMCB.b<  0) YMCB.b=  0;
        if (YMCB.b>min) YMCB.b=min;
    }
    YMCB.m-=YMCB.b;
    YMCB.c-=YMCB.b;
    YMCB.y-=YMCB.b;
    return(&YMCB);
}
```

リスト2 hcescv2i.s

```
            .include        hcescv2i.mac

            .xdef           _HCSET_SIKII
            .xdef           _HCSET_DF
            .xdef           _HCSET_CF
            .xdef           _HCESCV2I

            .xdef           _GETRGB_VRAM
            .xdef           _GETRGB_RAM2B
            .xdef           _GETRGB_RAM3B
            .xdef           _RGBTOYMCB

            .offset         0
HS_x                ds.w            1
HS_y                ds.w            1
HS_hx               ds.w            1
HS_hy               ds.w            1
HS_vx               ds.w            1
HS_vy               ds.w            1
HS_h_size   ds.w            1
HS_v_size   ds.w            1
HS_h_zoom   ds.w            1
HS_v_zoom   ds.w            1
HS_data             ds.l            1
HS_ff_flag  ds.w            1
HS_xs               ds.w            1
HS_ys               ds.w            1

            .offset         4
arg1                ds.l            1
arg2                ds.l            1
arg3                ds.l            1
arg4                ds.l            1
arg5                ds.l            1
arg6                ds.l            1
arg7                ds.l            1
arg8                ds.l            1

MCPY:       .macro          uar1,uar2,udr
            .local          loop
            lsr.w           #2,udr
            sub.w           #1,udr
loop:       move.l          (uar1)+,(uar2)+
            dbra.w          udr,loop
            .endm

SPREAD_J:   .macro
            .local          bse,mse,yse,cse

            add.w           d3,d3
            add.w           d4,d4
            add.w           d5,d5
            add.w           d6,d6

            move.w          (a0),d2
            cmp.w           SIKII_B,d2
            blt             bse
            move.w          SIKII_B,d1
            sub.w           d1,(a0)
            sub.w           d1,(a1)
            or.w            #1,d3
            bra             cse

bse:        move.w          2(a0),d2
            cmp.w           SIKII_M,d2
            blt             mse
            move.w          SIKII_M,d1
            sub.w           d1,2(a0)
            sub.w           d1,2(a1)
            or.w            #1,d4

mse:        move.w          4(a0),d2
            cmp.w           SIKII_Y,d2
            blt             yse
            move.w          SIKII_Y,d1
            sub.w           d1,4(a0)
            sub.w           d1,4(a1)
            or.w            #1,d5

yse:        move.w          6(a0),d2
            cmp.w           SIKII_C,d2
            blt             cse
            move.w          SIKII_C,d1
            sub.w           d1,6(a0)
            sub.w           d1,6(a1)
            or.w            #1,d6
cse:
            .endm

SPREAD_L:   .macro          udr
            .local          skip
            move.w          (a0),d2
skip:       asr.w           #1,d2
            add.w           d2,8(a0)
            asr.w           #1,d2
*           add.w           d2,-8(a1)
            asr.w           #1,d2
            add.w           d2,(a1)
            add.w           d2,8(a1)
            adda.w          #2,a0
            adda.w          #2,a1
            .endm

SPREAD_M    .macro          udr
            .local          skip
            move.w          (a0),d2
skip:       asr.w           #1,d2
            add.w           d2,8(a0)
            asr.w           #1,d2
            add.w           d2,-8(a1)
            asr.w           #1,d2
            add.w           d2,(a1)
            add.w           d2,8(a1)
            adda.w          #2,a0
            adda.w          #2,a1
            .endm

SPREAD_R    .macro          udr
            .local          skip
            move.w          (a0),d2
skip:       asr.w           #1,d2
*           add.w           d2,8(a0)
            asr.w           #1,d2
            add.w           d2,-8(a1)
            asr.w           #1,d2
            add.w           d2,(a1)
*           add.w           d2,8(a1)
            adda.w          #2,a0
            adda.w          #2,a1
            .endm

            .text
            .even

SIKII_B:    .dc.w           $0e00
SIKII_M:    .dc.w           $1200
SIKII_C:    .dc.w           $1200
SIKII_Y:    .dc.w           $1200
getrgb:             .dc.l           _GETRGB_VRAM
rgbtoymcb:  .dc.l           _RGBTOYMCB

_HCSET_SIKII: movem.l       a0-a1,-(sp)
            movea.l         arg1+8(sp),a0
            movea.l         #SIKII_B,a1
            move.w          (a0)+,(a1)+
            move.w          (a0)+,(a1)+
            move.w          (a0)+,(a1)+
            move.w          (a0)+,(a1)+
            movem.l         (sp)+,a0-a1
            rts

_HCSET_DF:  move.l          arg1(sp),getrgb
            rts

_HCSET_CF:  move.l          arg1(sp),rgbtoymcb
            rts

_HCESCV2I:  movem.l         d1-d7/a0-a6,-(sp)
            movea.l         arg1+56(sp),a0

            moveq.l         #_INIT_PRN,d0
            move.w          #$ffff,d1
            trap            #iocs
            cmp.l           #$20,d0
            bne             hcrtn_error1

            move.w          HS_h_size(a0),d0
            mulu.w          HS_h_zoom(a0),d0
            cmp.l           #$2000,d0
            bge             hcrtn_error3
            move.w          d0,hs_hz
            lsl.w           #2,d0
            move.w          d0,hs_hz_4
            add.w           d0,d0
            move.w          d0,hs_hz_4_2
            move.w          HS_v_zoom(a0),d1
            add.w           #3,d1
            mulu.w          d1,d0
            move.w          HS_h_size(a0),d1
            mulu.w          HS_h_zoom(a0),d1
            add.w           #15,d1
            and.w           #$fff0,d1
            move.w          d1,hs_hz_m
            lsr.w           #3,d1
            move.w          d1,hs_hz_b
            lsl.l           #2,d1
            add.l           d1,d0
            move.l          d0,-(sp)        * (2*512*4*XZ*(YZ+3)+512/8*XZ*4)
,-(sp)
            DOS             _MALLOC
            addq.l          #4,sp
            tst.l           d0
            blt             hcrtn_error2
            movea.l         d0,a6

*           moveq.l         #_B_SUPER,d0
*           movea.l         #0,a1
*           trap            #iocs
*           move.l          d0,-(sp)

```

▶10月に漢検2級，11月に英検準1級，1月にセンター試験，2月に大学受験……。いろいろありすぎて大変だけど，がんばらねば。　今井　佑(18)東京都

```
                movea.l         a6,a1
                move.w          HS_v_zoom(a0),d0
                add.w           #3,d0
                mulu.w          hs_hz_4_2,d0
                adda.l          d0,a1
                movea.l         a6,a2
                move.w          HS_v_zoom(a0),d0
                add.w           #1,d0
                mulu.w          hs_hz_4_2,d0
                adda.l          d0,a2
                movea.l         a6,a3
                move.w          HS_v_zoom(a0),d0
                add.w           #2,d0
                mulu.w          hs_hz_4_2,d0
                adda.l          d0,a3
                MJ_INZ
                MJ_OPT0
                movem.l         d0-d7/a0-a6,-(sp)
                move.l          a0,-(sp)
                move.l          #1,-(sp)
                move.l          getrgb,a0
                jsr             (a0)
                adda.w          #8,sp
                movem.l         (sp)+,d0-d7/a0-a6
                jsr             GETLINE
                movea.l         a6,a5
                movea.l         a3,a4
                move.w          hs_hz_4_2,d7
                MCPY            a4,a5,d7
                move.w          HS_v_size(a0),d0
                sub.w           #2,d0
                blt             hclpend
hclp:           move.w          d0,-(sp)
                move.w          HS_vx(a0),d1
                add.w           d1,HS_x(a0)
                move.w          HS_vy(a0),d1
                add.w           d1,HS_y(a0)
                movem.l         d0-d7/a0-a6,-(sp)
                move.l          a0,-(sp)
                move.l          #2,-(sp)
                move.l          getrgb,a0
                jsr             (a0)
                adda.w          #8,sp
                movem.l         (sp)+,d0-d7/a0-a6
                exg.l           a2,a3
                jsr             GETLINE
                movea.l         a6,a4
                jsr             SMOOTH
                movea.l         a3,a4
                movea.l         a6,a5
                move.w          HS_v_zoom(a0),d0
                mulu.w          hs_hz_4_2,d0
                adda.l          d0,a5
                move.w          hs_hz_4_2,d7
                MCPY            a4,a5,d7
                movea.l         a6,a4
                jsr             MJLINEOUT
                movea.l         a6,a4
                move.w          HS_v_zoom(a0),d0
                mulu.w          hs_hz_4_2,d0
                adda.l          d0,a4
                movea.l         a6,a5
                move.w          hs_hz_4_2,d7
                MCPY            a4,a5,d7
                move.w          (sp)+,d0
                dbra.w          d0,hclp
hclpend:        move.w          HS_v_zoom(a0),d0
                sub.w           #1,d0
                movea.l         a6,a5
                clr.l           d7
                move.w          hs_hz_4_2,d7
                adda.l          d7,a5
hclp2:          movea.l         a3,a4
                clr.l           d7
                move.w          hs_hz_4_2,d7
                MCPY            a4,a5,d7
                dbra.w          d0,hclp2
                jsr             MJLINEOUT
                movem.l         d0-d7/a0-a6,-(sp)
                move.l          a0,-(sp)
                move.l          #3,-(sp)
                move.l          getrgb,a0
                jsr             (a0)
                adda.w          #8,sp
                movem.l         (sp)+,d0-d7/a0-a6
                MJ_INZ
                move.l          a6,-(sp)
                DOS             _MFREE
                addq.l          #4,sp
*               move.l          (sp)+,d0
*               cmp.l           #1,d0
*               beq             rcv_um_skip
*               movea.l         d0,a1
*               moveq.l         #_B_SUPER,d0
*               trap            #iocs
rcv_um_skip:
                tst.w           HS_ff_flag(a0)
                beq             ff_skip
                MJ_FF
ff_skip:        moveq.l         #0,d0
                movem.l         (sp)+,d1-d7/a0-a6
                rts
hcrtn_error1: moveq.l           #-1,d0
                movem.l         (sp)+,d1-d7/a0-a6
                rts
hcrtn_error2: moveq.l           #-2,d0
                movem.l         (sp)+,d1-d7/a0-a6
                rts
hcrtn_error3: moveq.l           #-3,d0
                movem.l         (sp)+,d1-d7/a0-a6
                rts
MJLINEOUT: movem.l              a0/a2-a5,-(sp)
                movea.l         a1,a2
                movea.l         a2,a3
                adda.w          hs_hz_b,a3
                movea.l         a3,a4
                adda.w          hs_hz_b,a4
                movea.l         a4,a5
                adda.w          hs_hz_b,a5
                move.w          HS_v_zoom(a0),d7
                sub.w           #1,d7
                movea.l         a6,a0
mol:
                move.w          d7,-(sp)
                jsr             SPREAD
                move.l          a1,-(sp)
                MJ_LINE_MCYB    a1,hs_hz_m
                move.l          (sp)+,a1
                move.w          (sp)+,d7
                dbra.w          d7,mol
                movem.l         (sp)+,a0/a2-a5
                rts
SPREAD:         movem.l         a1-a5,-(sp)
                movea.l         a0,a1
                clr.l           d0
                move.w          hs_hz_4_2,d0
                adda.l          d0,a1
                move.w          #16,d7
                SPREAD_J
                SPREAD_L        d3
                SPREAD_L        d4
                SPREAD_L        d5
                SPREAD_L        d6
                sub.w           #1,d7
                move.w          hs_hz,d0
                sub.w           #3,d0
spl1:
                SPREAD_J
                SPREAD_M        d3
                SPREAD_M        d4
                SPREAD_M        d5
                SPREAD_M        d6
                sub.w           #1,d7
                bgt             sps0
                move.w          d3,(a2)+
                move.w          d4,(a3)+
                move.w          d5,(a4)+
                move.w          d6,(a5)+
                move.w          #16,d7
sps0:           dbra.w          d0,spl1
                SPREAD_J
                SPREAD_R        d3
                SPREAD_R        d4
                SPREAD_R        d5
                SPREAD_R        d6
sps1:           sub.w           #1,d7
                ble             sps2
                add.w           d3,d3
                add.w           d4,d4
                add.w           d5,d5
                add.w           d6,d6
                bra             sps1
sps2:           move.w          d3,(a2)+
                move.w          d4,(a3)+
                move.w          d5,(a4)+
                move.w          d6,(a5)+
                move.w          #16,d7
                movem.l         (sp)+,a1-a5
                rts
SMOOTH:         movem.l         a2-a3,-(sp)
                clr.l           d3
                move.w          hs_hz_4_2,d3
                move.w          hs_hz_4,d7
                sub.w           #1,d7
sl0:            movea.l         a4,a5
                adda.w          #2,a4
                move.w          (a2)+,d4
                move.w          (a3)+,d5
                sub.w           d4,d5
                ext.l           d5
                divs.w          HS_v_zoom(a0),d5
                move.w          HS_v_zoom(a0),d6
                sub.w           #2,d6
                blt             sl1end
sl1:            add.w           d5,d4
                adda.l          d3,a5
                move.w          d4,(a5)
                dbra.w          d6,sl1
sl1end:         dbra.w          d7,sl0
                movem.l         (sp)+,a2-a3
                rts
GETLINE:        move.l          a3,-(sp)
                move.w          HS_h_size(a0),d6
                sub.w           #1,d6
                move.l          a0,-(sp)
                jsr             GETDOT
                move.w          (a0)+,d0
                move.w          (a0)+,d1
                move.w          (a0)+,d2
                move.w          (a0)+,d3
                move.l          (sp)+,a0
                sub.w           #1,d6
                blt             glend
ryl0:
                move.w          d6,-(sp)
                move.l          a0,-(sp)
                jsr             GETDOT
                move.w          (a0)+,d4
                move.w          (a0)+,d5
                move.w          (a0)+,d6
                move.w          (a0)+,d7
                move.l          (sp)+,a0
                movem.w         d4-d7,-(sp)
                sub.w           d0,d4
                sub.w           d1,d5
                sub.w           d2,d6
                sub.w           d3,d7
                ext.l           d4
                ext.l           d5
                ext.l           d6
                ext.l           d7
                divs.w          HS_h_zoom(a0),d4
                divs.w          HS_h_zoom(a0),d5
                divs.w          HS_h_zoom(a0),d6
                divs.w          HS_h_zoom(a0),d7
                movem.w         d4-d7,-(sp)
                move.w          HS_h_zoom(a0),d7
```

▶ 9月号から値段が上がったんですね。レジで思わず「エッ？」って聞き返してしまいました。
後藤　恭利(23)福岡県

```
                sub.w           #1,d7
ryl1:           move.w          d0,(a3)+
                move.w          d1,(a3)+
                move.w          d2,(a3)+
                move.w          d3,(a3)+
                add.w           (sp),d0
                add.w           2(sp),d1
                add.w           4(sp),d2
                add.w           6(sp),d3
                dbra.w          d7,ryl1
                adda.w          #8,sp
                movem.w         (sp)+,d0-d3
                move.w          (sp)+,d6
                dbra.w          d6,ryl0

glend:
                move.w          HS_h_zoom(a0),d7
                sub.w           #1,d7
ryl2:           move.w          d0,(a3)+
                move.w          d1,(a3)+
                move.w          d2,(a3)+
                move.w          d3,(a3)+
                dbra.w          d7,ryl2
glrtn:
                move.l          (sp)+,a3
                rts

GETDOT:
                movem.l         d0-d7/a1-a6,-(sp)
                move.l          a0,-(sp)

                move.l          a0,-(sp)
                move.l          #0,-(sp)
                movea.l         getrgb,a0
                jsr             (a0)
                adda.w          #8,sp
                move.l          d0,-(sp)
                movea.l         rgbtoymcb,a0
                jsr             (a0)
                adda.w          #4,sp

                move.l          (sp)+,a0
                move.w          HS_hx(a0),d1
                add.w           d1,HS_x(a0)
                move.w          HS_hy(a0),d1
                add.w           d1,HS_y(a0)
                movea.l         d0,a0
                movem.l         (sp)+,d0-d7/a1-a6
                rts

hs_hz:          .dc.w           0
hs_hz_4:        .dc.w           0
hs_hz_4_2:      .dc.w           0
hs_hz_m:        .dc.w           0
hs_hz_b:        .dc.w           0

_GETRGB_VRAM:
                move.l          #$e82400,d0
                movea.l         d0,a1
                move.w          (a1),d1
                and.w           #7,d1
                move.l          arg1(sp),d0
                tst.w           d0
                bne             grvrtn
                movea.l         arg2(sp),a0
                clr.l           d0
                move.w          HS_y(a0),d0
                lsl.w           #2,d0
                lsl.l           #8,d0
                btst.l          #2,d1
                beq             grss0
                add.l           d0,d0
grss0:          clr.l           d2
                move.w          HS_x(a0),d2
                add.l           d2,d0
                add.l           d2,d0
                add.l           HS_data(a0),d0
                movea.l         d0,a1
                move.w          (a1),d0
                cmp.w           #3,d1
                beq             grss1
                and.l           #$000000ff,d0
                add.w           d0,d0
                add.l           #$e82000,d0
                movea.l         d0,a1
                move.w          (a1),d0
grss1:          move.w          d0,d1
                and.w           #$f800,d1
                lsr.w           #5,d1
                move.w          d0,d2
                and.w           #$07c0,d2
*               lsr.w           #1,d2
                move.w          d0,d3
                and.w           #$003e,d3
                lsl.w           #5,d3

                move.w          d1,-(sp)
                lsr.w           #5,d1
                or.w            (sp),d1
                lsr.w           #5,d1
                or.w            (sp)+,d1
                move.w          d2,-(sp)
                lsr.w           #5,d2
                or.w            (sp),d2
                lsr.w           #5,d2
                or.w            (sp)+,d2
                move.w          d3,-(sp)
                or.w            (sp),d3
                lsr.w           #5,d3
                lsr.w           #5,d3
                or.w            (sp)+,d3

                movea.l         #GRB,a0
                move.w          d1,(a0)
                move.w          d2,2(a0)
                move.w          d3,4(a0)
                move.l          a0,d0
grvrtn:
                rts

_GETRGB_RAM2B: move.l           arg1(sp),d0
                movea.l         arg2(sp),a0
                tst.w           d0
                beq             grr2com0
                cmp.w           #2,d0
                beq             grr2rtn
                bgt             grr2com3
                move.w          HS_xs(a0),d0
                mulu.w          HS_ys(a0),d0
                add.l           d0,d0
                move.l          d0,-(sp)
                DOS             _MALLOC
                addq.l          #4,sp
                tst.l           d0
                blt             grr2rtn
                move.l          d0,HS_data(a0)
                clr.l           d0
                rts
grr2com3:       move.l          HS_data(a0),-(sp)
                DOS             _MFREE
                addq.l          #4,sp
                rts
grr2com0:       move.w          HS_y(a0),d0
                mulu.w          HS_xs(a0),d0
                clr.l           d1
                move.w          HS_x(a0),d1
                add.l           d1,d0
                add.l           d0,d0
                add.l           HS_data(a0),d0
                movea.l         d0,a1
                move.w          (a1)+,d0
                move.w          d0,d1
                and.w           #$f800,d1
                lsr.w           #5,d1
                move.w          d0,d2
                and.w           #$07c0,d2
                move.w          d0,d3
                and.w           #$003e,d3
                lsl.w           #5,d3

                move.w          d1,-(sp)
                lsr.w           #5,d1
                or.w            (sp),d1
                lsr.w           #5,d1
                or.w            (sp)+,d1
                move.w          d2,-(sp)
                lsr.w           #5,d2
                or.w            (sp),d2
                lsr.w           #5,d2
                or.w            (sp)+,d2
                move.w          d3,-(sp)
                or.w            (sp),d3
                lsr.w           #5,d3
                lsr.w           #5,d3
                or.w            (sp)+,d3

                movea.l         #GRB,a0
                move.w          d1,(a0)
                move.w          d2,2(a0)
                move.w          d3,4(a0)
                move.l          a0,d0
grr2rtn:        rts

_GETRGB_RAM3B: move.l           arg1(sp),d0
                movea.l         arg2(sp),a0
                tst.w           d0
                beq             grr3com0
                cmp.w           #2,d0
                beq             grr3rtn
                bgt             grr3com3
                move.w          HS_xs(a0),d0
                mulu.w          HS_ys(a0),d0
                move.l          d0,d1
                add.l           d0,d0
                add.l           d1,d0
                move.l          d0,-(sp)
                DOS             _MALLOC
                addq.l          #4,sp
                tst.l           d0
                blt             grr3rtn
                move.l          d0,HS_data(a0)
                clr.l           d0
                rts
grr3com3:       move.l          HS_data(a0),-(sp)
                DOS             _MFREE
                addq.l          #4,sp
                rts
grr3com0:       move.w          HS_y(a0),d0
                mulu.w          HS_xs(a0),d0
                clr.l           d1
                move.w          HS_x(a0),d1
                add.l           d1,d0
                move.l          d0,d1
                add.l           d1,d0
                add.l           d1,d0
                add.l           HS_data(a0),d0
                movea.l         d0,a1
                clr.w           d1
                clr.w           d2
                clr.w           d3
                move.b          (a1)+,d2
                move.b          (a1)+,d1
                move.b          (a1)+,d3
                lsl.w           #3,d1
                lsl.w           #3,d2
                lsl.w           #3,d3

                move.w          d1,-(sp)
                lsr.w           #8,d1
                or.w            (sp)+,d1
                move.w          d2,-(sp)
                lsr.w           #8,d2
                or.w            (sp)+,d2
                move.w          d3,-(sp)
                lsr.w           #8,d3
                or.w            (sp)+,d3

                movea.l         #GRB,a0
                move.w          d1,(a0)
                move.w          d2,2(a0)
                move.w          d3,4(a0)
                move.l          a0,d0
grr3rtn:        rts

GRB:                    .dc.w           0
                .dc.w           0
                .dc.w           0

_RGBTOYMCB:
                move.l          arg1(sp),a0
                move.w          (a0)+,d1
                move.w          (a0)+,d2
                move.w          (a0)+,d3
                eor.w           #$07ff,d1
                eor.w           #$07ff,d2
                eor.w           #$07ff,d3

                move.w          d1,d4
                move.w          d2,d5
                cmp.w           d4,d5
                bge             s0
                exg.l           d4,d5
s0:             cmp.w           d4,d3
                bge             s1
```

▶ X68000 PROが互換機だなんて同じメーカーから出ているのに納得できん。
山脇　彰(20)福岡県

```
                move.w          d3,d4
                bra             s2
s1:             cmp.w           d5,d3
                ble             s2
                move.w          d3,d5
s2:             tst.w           d5
                bgt             s3
                move.w          #1,d5
s3:
                clr.w           d0
                cmp.w           #$02ff,d4
                ble             s5
                move.w          d4,d0
                sub.w           #$02ff,d0
                mulu.w          #$7fff,d0
                divu.w          #$7d00,d0
*               clr.w           d0
*               cmp.w           #$01ff,d5
*               ble             s5
*               move.w          d5,d6
*               sub.w           d4,d6
*               cmp.w           #$03ff,d6
*               bge             s5

                move.w          d4,d0
                mulu.w          d4,d0
                divu.w          #$07ff,d0
                mulu.w          d4,d0
                divu.w          d5,d0
s5:
                sub.w           d0,d1
                sub.w           d0,d2
                sub.w           d0,d3
                movea.l         #BMCY,a0
                move.w          d0,(a0)
                move.w          d1,2(a0)
                move.w          d2,4(a0)
                move.w          d3,6(a0)
                move.l          a0,d0
                rts

BMCY:           .dc.w           0
                .dc.w           0
                .dc.w           0
                .dc.w           0

                .end
```

リスト3 hcescv2i.mac

```
            .include        iocscall.mac
            .include        doscall.mac

iocs:               .equ            $0f

mj_b:               .equ            $00
mj_m:               .equ            $01
mj_c:               .equ            $02
mj_y:               .equ            $04

OUTLPT_CALL:        .macro
            moveq.l         #_OUTLPT,d0
            trap            #iocs
            .endm

OUTLPT_C:   .macro          agl
            move.l          agl,d1
            OUTLPT_CALL
            .endm

OUTLPT_S:   .macro          agl
            move.w          agl,d1
            OUTLPT_CALL
            lsr.w           #8,d1
            OUTLPT_CALL
            .endm

OUTLPT_M:   .macro          agl,uar
            .local          loop
            move.l          agl,d2
            sub.w           #1,d2
loop:       move.b          (uar)+,d1
            OUTLPT_CALL
            dbra.w          d2,loop
            .endm

MJ_INZ:     .macro
            OUTLPT_C        #$1b            * プリンタの初期化
            OUTLPT_C        #$40
            .endm

MJ_OPT0:    .macro
            OUTLPT_C        #$1b            * グラフィックモード移行
            OUTLPT_C        #$28
            OUTLPT_C        #$47
            OUTLPT_C        #$01
            OUTLPT_C        #$00
            OUTLPT_C        #$01

            OUTLPT_C        #$1b            * マイクロウィーブモード
            OUTLPT_C        #$28
            OUTLPT_C        #$69
            OUTLPT_C        #$01
            OUTLPT_C        #$00
            OUTLPT_C        #$01

            OUTLPT_C        #$1b            * ユニット
            OUTLPT_C        #$28
            OUTLPT_C        #$55
            OUTLPT_C        #$01
            OUTLPT_C        #$00
            OUTLPT_C        #$05
            .endm

MJ_COLOR:   .macro          color
            OUTLPT_C        #$1b            * カラー
            OUTLPT_C        #$72
            OUTLPT_C        color
            .endm

MJ_LUSTER:  .macro
            OUTLPT_C        #$1b            * ラスターグラフィックモード(縦360*横720dpi)
            OUTLPT_C        #$2e
            OUTLPT_C        #$00
            OUTLPT_C        #$05
            OUTLPT_C        #$05
            OUTLPT_C        #$01
            .endm

MJ_HR:      .macro
            OUTLPT_C        #$0d
            .endm

MJ_CR720:   .macro
            OUTLPT_C        #$1b            * 1/720インチ改行
            OUTLPT_C        #$28
            OUTLPT_C        #$76
            OUTLPT_C        #$02
            OUTLPT_C        #$00
            OUTLPT_C        #$01
            OUTLPT_C        #$00
            .endm

MJ_FF:      .macro
            OUTLPT_C        #$0c
            .endm

MJ_LINE:    .macro          uar,color,cnt
            MJ_COLOR        color
            MJ_LUSTER
            OUTLPT_S        cnt
            move.w          cnt,d0
            add.w           #7,d0
            lsr.w           #3,d0
            OUTLPT_M        d0,uar
            MJ_HR
            .endm

MJ_LINE_MCYB: .macro        uar,cnt
            MJ_LINE         uar,#mj_b,cnt
            MJ_LINE         uar,#mj_m,cnt
            MJ_LINE         uar,#mj_c,cnt
            MJ_LINE         uar,#mj_y,cnt
            MJ_CR720
            .endm
```

リスト4 hcescv2i.h

```
#define         RGB_MAX         0x07ff  /*内部で処理するRGB各色の輝度の最大値*/

#define         A4H             0x1680  /*A4サイズに印刷可能な水平方向の最大ドット数*/
#define         A4V             0x1e00  /*A4サイズに印刷可能な垂直方向の最大ドット数*/
                                        /*(正確には若干小さい)*/
struct      hcstat  {
            short           x;      /*元画像の印刷位置*/
            short           y;
            short           hx;     /*ヘッド水平方向移動時の元画像の印刷位置の移動量*/
            short           hy;
            short           vx;     /*ヘッド垂直方向移動時の元画像の印刷位置の移動量*/
            short           vy;
            short           h_size; /*用紙の水平方向に印刷する元画像のドット数*/
            short           v_size; /*用紙の垂直方向に印刷する元画像のドット数*/
            short           h_zoom; /*元画像の1ドットに対するプリンタのドット数*/
            short           v_zoom;
            void            *data;  /*画像データアドレス*/
            short           ff_flag;/*印刷後の紙送り*/
            short           xs;     /*元画像のドット数*/
            short           ys;

            short           x1;     /*元画像の印刷範囲*/
            short           y1;
            short           x2;
            short           y2;
            short           wxs;    /*元画像の印刷対象サイズ*/
            short           wys;
            unsigned long   color;  /*元画像の色数*/
            short           hv_flag;/*元画像と用紙との向き*/
            short           ph;     /*用紙のサイズ*/
            short           pv;
};

struct      rgbstat {
            short           green;  /*緑の輝度*/
            short           red;    /*赤の輝度*/
            short           blue;   /*青の輝度*/
    };

struct ymcbstat     {
            short           b;      /*    黒の濃さ*/
            short           m;      /*マゼンダの濃さ*/
            short           c;      /*  シアンの濃さ*/
            short           y;      /*   黄色の濃さ*/
    };

short       HCESCV2I(struct hcstat *);                              /*ハードコピーのメイン*/

struct      rgbstat *GETRGB_VRAM(int,struct hcstat *);              /*VRAMの印刷*/
struct      rgbstat *GETRGB_RAM2B(int,struct hcstat *);             /*RGB2バイトデータの印刷*/
struct      rgbstat *GETRGB_RAM3B(int,struct hcstat *);             /*RGB3バイトデータの印刷*/

struct ymcbstat     *RGBTOYMCB(struct rgbstat *);                   /*池田による四色分解*/
```

リスト5 makefile

```
hcescv2i.x: ¥
    main.o ¥
    hcescv2i.o
    cc /Gh8k /Fx$@ $^ clib.l doslib.l iocslib.l baslib.l
main.o: main.c
    cc /Fc /Fo$@ $<
hcescv2i.o: hcescv2i.s hcescv2i.mac
    as /o$@ $<
```

▶先日，友人Kが過去に六本木パソコンを所有していたという事実が発覚しました。いったいなにが彼をそうさせたのか？　さまざまな憶測が飛び交っています。ところで今年の特集タイトルは，なぜ横文字なのか？　怪しい。　伊藤　義博(25)東京都

ESPER MACHとMACH JETを使いこなす

MACH JETの使い方

HCESCV2Xを作った瀧です。

1年前に私がMACH JET用印刷プログラムを作っていたときには，いろいろ苦戦してはみたものの，結果的に720×360dpiでしか印刷ができませんでした。しかし，その後，縦方向を720dpiで印刷制御する方法がわかりましたので報告します。

さらにその後，MJ5000C，MJ900Cが発売されました。この2つにはESPER MACHモードという，ドットをさらに細かくするモードが装備されていました（dpi自体は変わらない）。

これらの制御方法もある程度わかったので，ここでエプソンMACH JET系カラープリンタでの印刷の手順をまとめて紹介してみたいと思います。

初期化処理

グラフィックデータを印刷するにはプリンタにどのようなモードで印字するのかを指定しなくてはなりません。すなわち初期化処理です。最初にプリンタに対して以下のような手順でコマンドを送ります。

1）初期化

　$1b，$40

プリンタを初期化します。

MJシリーズのプリンタは内蔵で少しだけプリンタバッファを持っていますが，もしもプリンタバッファにデータが残っているのならば，このコマンドでそこにあるデータをフラッシュします。

2）グラフィックモード移行

　$1b，$28，$47，$01，$00，$01

ラスタグラフィック印刷モードに移行します。

3）基本ユニット設定

　$1B，$28，$55，$01，$00，m

dpi値を設定します。m/3600インチ単位に設定可能。たとえば，720dpiにしたいのならば5を指定します。

4）マイクロウィーブ設定

　$1b，$28，$69，$01，$00，$01

ラインプリンタの宿命である，ヘッド幅分の横縞（バンディング）を消すための設定です。これでかなりバンディングが目立たなくなります。

5）ドットサイズ変更

　$1B，$28，$65，$02，$00，$00，n

MACH JETにはこの命令はありません。

n=1で微小ドットモード（ESPER MACHモード）に設定されます。微小ドットモードにすると，全体的に色合いが薄くなるので，注意が必要です。なお，MJ700V2Cなどでは指定しても無視されます。

1ラスタループの動作

次に，実際にグラフィックデータ本体を順次送っていきます。データは横1本線（ラスタ）ずつ以下のような手順で送っていきます。このループを希望するサイズ分だけ繰り返せばよいわけです。

1）左マージン移動

　$1b

ヘッドをいちばん左に移動します。

2）カラー設定

　$1b，$72，$c

色を設定。

　黒　　＝0
　マゼンタ＝1
　シアン　＝2
　イエロー＝4

になります。

3）ラスタグラフィック印刷

　$1b，$2e，$00，$05，$05，$01，・・・・

1ラスタ分のグラフィックデータを印刷します。

4）CRを出力

5）2)まで3色分ループ

色の順序はブラック，マゼンタ，シアン，イエローの順。色の薄いものの上に濃いものを書くと色が潰れてしまうからです。

6）1ライン下に移動

　$1b，$28，$76，$02，$00，$01，$00

＊　＊　＊

これで，印刷ができるはずです。そのほか，MJ700V2Cでは処理が遅いようで，実用的ではないとかいう話もあります。印刷自体はMJ800CよりもMJ700V2Cのほうが，インクの違いからか綺麗になるという話もあります。

微小モードの追加

今回の投稿プログラムHCESCV21ですが，HCESCV21.MAC中の初期化ルーチンに，微小モード設定を加えると，MJ5000C，MJ900Cでさらに綺麗な印刷ができるようになります。ただし，このままでは色が薄くなるので，色を濃い目に印刷するように各自で設定しておいてください。

微小モードの設定部分はリスト1のようになります。これをソースプログラムに加えてください。

微小モードが追加されたことによって，カラー印字の可能性はかなり拡大されたことがわかります。実際にエプソンのドライバがなにをやっているのかはよくわからないのですが，色表現，階調表現が格段に上がることが考えられます。

従来はドットを打つ・打たないの2つの選択肢から色を決定していましたので，使用できる色は最大でも2の4乗＝16色でした（これは黒も考慮しているが，実際には混色として使われることはほとんどないと思われる）。

それが，微小モード時のインク噴出量が通常時の1/2とした場合（正確な量は不明），3の4乗＝81色，黒を除いても27色の表現が理論上可能になるわけですから。

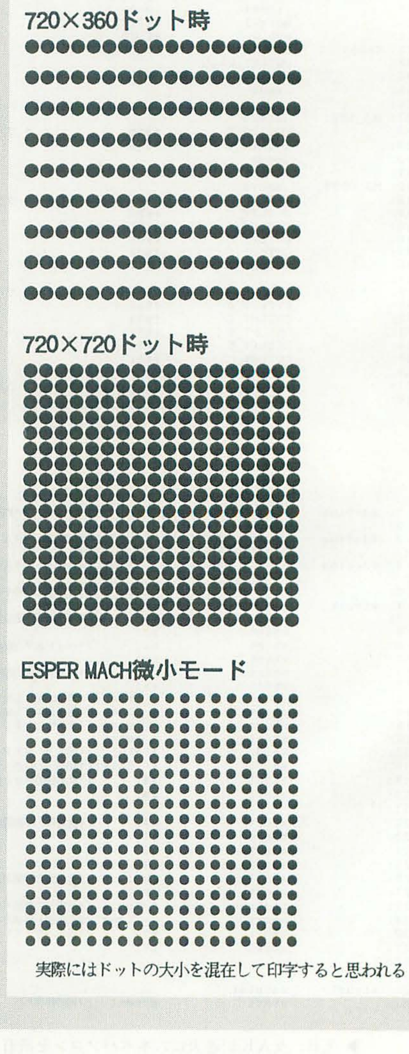
720dpiという解像度と低価格さでMACH JETシリーズも大躍進を続けています。ほかに気になるプリンタとしてはアルプス電気の熱転写プリンタMD2000などがあるのですが，パラレルインタフェイス版が出てくるのが10月以降ということで今回は接続試験できませんでした。今後手に入り次第確認してみたいと思います。

（瀧　康史）

リスト1　微小モードへの移行

```
OUTLPT_C    #$1b    ＊微小モード移行
OUTLPT_C    #$28
OUTLPT_C    #$65
OUTLPT_C    #$02
OUTLPT_C    #$00
OUTLPT_C    #$00
OUTLPT_C    #$01
```

図1　MACH JET印字解像度の概念図



実際にはドットの大小を混在して印字すると思われる

(で)のショートプロぱーてぃ――その73

真実に勝るモノなし！

Komura Satoshi 古村 聡

やはりフィクションはフィクション，現実のリアルさにはかないませんがな，とセミの墓場に学んだ(で)氏。今月はゲームが1本にツールが2本です。ツールは結構本格的なものなので気合を入れて使いましょう。

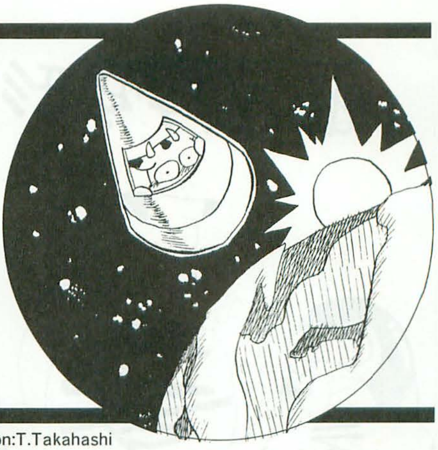


illustration:T.Takahashi

いよいよ本格的な秋がやってきました。ところで，最近せっかく猛暑が終わって涼しくなり，ホッとひと息ついたのにさらに涼しくなる体験をしてしまいました。

場所は引越し前の家。前に住んでいたところはすごい場所で，鉄筋築5年のクセに隙間風ヒューヒュー(おばけではなく手抜き工事です。どう考えても)なんてのもありましたが，偶然見たのが世にも恐ろしい光景だったのです。

この話は，ひと月くらい遡ったときから始まります。どういうわけだか，家の前にはセミの死体がコロコロ転がっていたんですね。日によっては4つも5つもです。朝，晩，家を出入りするたびに落ちてる場所も数も違うので，毎日違うセミが死んでいるのだろうな，とは思っていたのですが，それにしては前日あったセミの死体はどこへ行ってしまったのか？　どこかにセミの墓場でもあるのか？　ちょっとだけ不思議に思っていたのです。

そして，引越し当日。家の前も掃除していこうかな，なんて思って，ふと家の前に水を流してみたら，水の流れが悪いんです。あれ？　と思ってなにげなく排水口を見てみたら，そこにあったのは……。

「セミの墓場だった！」

排水口いっぱい，ぎっしりとセミの死体がつまっておりました。うぇえ。

うーむ，真実に勝るホラーなどありませんね，まったくもって。ううっ，思い出すだけでも吐き気が。

アポロチョコの楕円飛行

ということで，ここしばらくはホラー映画を観る気にはならないので「アポロ13」でも観にいってまいります。やっぱ真実に勝るシナリオはないやねえ(元シナリオライターの私の立場は？)。と強引な展開で今月の1本目のプログラムは，月面軌道周回ゲームAPOLLO.Xです。どうぞ！

APOLLO.X for X680x0
(要GCCコンパイラ，ジョイスティック)
大阪府　小枝直隆

リスト1を打ち込んで，以下のようにしてGCCでコンパイルしてください。

```
A＞GCC APOLLO.C -O -fomit-frame-pointer -finline-functions -fforce-mem -liocs
```

無事，APOLLO.Xという名前で実行ファイルができたら，

```
A>APOLLO
```

で実行です。画面には，青い地球に黄色い月が表示され，地球の周りを三角のロケットが周回軌道を回ってます。ジョイスティック1を使って，ロケットをなるべく安全に，なるべく少ない燃料で月旅行を成功させましょう。スティックを上に押すことで加速，下で減速，左右でそれぞれ左右に加速がつきます。

そして，195行のclk＝10の値を変えることによって処理速度を変えられます。たとえば16MHzのX68000 XVIなら16，24MHzのX68000 REDZONEなら24にするとほぼ同じようなスピードになります(X68030だと40ぐらいかな)。

うーむ，なかなか難しいざんすね。実はこのゲーム，ゲームオーバーってものが存在しません。とりあえず月と地球の周回軌道に乗せることが目的なんですけど，なかなか行って帰ってくるだけでも結構難しいんです。なぜか月旅行が恒星間片道旅行になっちゃうし(はやい話が，あさってに向かってGO！　になる)。

ちなみに，作者の小枝さんが投稿原稿に書いてくれた軌道に乗るコツ，なんでありますが，

「だいたい放っておくと楕円軌道に乗れます。その状況で近日点(楕円軌道の一番惑星に近い点)でジョイスティックを下に倒して減速，遠日点でジョイスティックを上にして加速するようにすれば真円に近くなるようです(経験則です)。逆にやればどんどん楕円になるので，うまくやるとちゃんと月旅行ができます」

ということだそうです。私がやると思いっきり月をかすめて「goto明後日」なんですけども(きっとよっぽどお星様の世界が好きなのね)。映画を観る前に，すでにジムラベル船長ってばすごいのね，と感動してしまったのでありました。

WYSIWYFかもしれないのだ

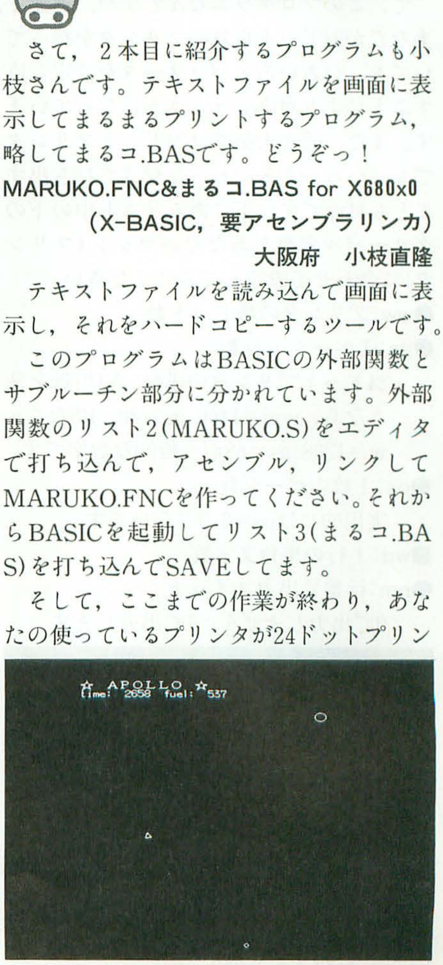
さて，2本目に紹介するプログラムも小枝さんです。テキストファイルを画面に表示してまるまるプリントするプログラム，略してまるコ.BASです。どうぞっ！

MARUKO.FNC&まるコ.BAS for X680x0
(X-BASIC，要アセンブラリンカ)
大阪府　小枝直隆

テキストファイルを読み込んで画面に表示し，それをハードコピーするツールです。

このプログラムはBASICの外部関数とサブルーチン部分に分かれています。外部関数のリスト2(MARUKO.S)をエディタで打ち込んで，アセンブル，リンクしてMARUKO.FNCを作ってください。それからBASICを起動してリスト3(まるコ.BAS)を打ち込んでSAVEしてます。

そして，ここまでの作業が終わり，あなたの使っているプリンタが24ドットプリン



APOLLO.C



タだったら，とりあえずRUNしてみましょう。「データファイル（フルパスで)」と表示されたら，印字したいファイル名を入れてください。続いて，印字フォーマットを聞いてきますので，数字で選んでください。設定が正しければ，画面にファイルの内容を文字のイメージを表示しながら，プリンタがプリントアウトしてくれます。

で，このプログラムなんですが，運悪く，あなたが24ドット以外のプリンタを持っていても，いろいろとパラメータを設定し直すことにより対応できるようになっています。また，設定次第では24ドットプリンタでも，いろいろと変わった形式で打ち出すことも可能です。下にあるリスト中の下のグローバル変数をあなたのマシン(プリンタ)に合わせて設定してみてください。

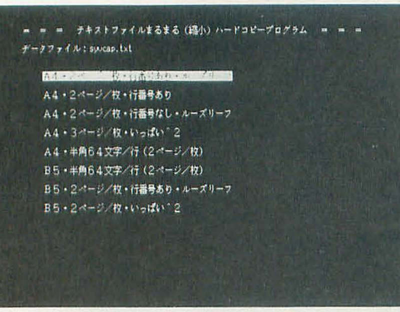
●ww:プリンタの横ドット数

●gs:1ページの行数
24ドットプリンタの場合，A4用紙を使うなら，ww=1400,gs=80。B5ならww=1208,gs=68が一般的な数字です

●ps:1枚のページ数
実用的にはps=2,3ぐらいです

●wd:1行の半角文字数

●nm:行番号出力スイッチ
0で出力しません，1で出力します



MARUKO.FNC&まるコ.BAS

●nf:行番号フォント
&H8000　半角文字
&HF000　上1/4角文字
&HF200　下1/4角文字

●ns:行番号ステップ
行番号が何文字ごとにつけられるかのパラメータです。ns=5なら，5,10,15行のときに行番号がつきます

●PZ:ページ出力モード
0で左づめ，1で中央，2で右づめ

●sh, sf:ヘッダ，フッダのスイッチ
0=出力しない　2=出力する

●pch:パンチ穴用横スペースドット数
0=なし　48が一般的

●poe:パンチ穴左右スイッチ
0=奇数枚目の左,偶数枚目の右にスペースを空ける
1=奇数枚目の右,偶数枚目の左にスペースを空ける

●st, ed:印字または.dviファイル作成開始，終了枚目
st<edに設定すると画面表示のみ，つまり「プレビューア」として働きます

●moe:印字またはファイル作成，奇数，偶数スイッチ
0=奇数のみ
1=偶数のみ
それ以外=奇数偶数両方

●ls:.dviファイル作成スイッチ
0=プリンタに直接印字出力，ファイルは作らない
1=ファイルを作成，プリンタには印字しない(デフォルトではドライブHのディレクトリ¥TXH_DVIの下にファイル名D_(no.).dviで作ります。1190行でこのディレクトリ名を決めていますので，違うディレクトリにする場合はここを変えてください。また，このファイルをコマンドライン上でLPTにコピーすると直接印字したものと同じ結果が得られます)


REFONT.C

おお，これは，8月号のショートプロで「もう一歩」のほうになってしまった小枝さんのtxread.fncの改良版なのですね。今度はBASICからかなり実用的なプログラムまでついていて，さすがにバッチシでありますね。竹本泉フォントなんか入れた日には文章がかわいいぞ～。わしの文章がなんだかかあいいよ～。うぷぷぷぷ。なんか妙におかしいぞ。今月，小枝さんは2本連チャンで掲載になってしまいましたが，これだけちゃんと作ってあれば2本掲載も当然ですね。もちのロン，タンヤオで1000点(意味不明)。

写像フォントで字を変えろ!

今月最後のプログラム。3本目は写像によってアウトランフォントを変形するプログラム，REFONT.Xです。どうぞっ。

REFONT.X for X680x0

(要XCコンパイラ)
新潟県　林直貴

「ちょっと前に，パソコンを使用していないときに，A4いっぱいに巨大な文字で印刷したよくわからない短文を垂らしていました。いくつかのバージョンをその日の気分で替えていましたが，フォントだけはもう少しぶっ飛んだものがないかと，常々思っていました。そこで作ってみたのが，この写像フォントコンバータREFONT.Xです。

書体倶楽部のアウトラインフォントは，頂点同士を直線で結ぶ簡単なものです。そこで，この頂点の座標をある種の写像関数で，別の領域に写像したらどうなるでしょう？　たとえば図A)のように写像したなら，きっと面白いフォントができることでしょう。絶対に，こんなフォントができるはずです(実際には，あまり長い直線は点の補間なども行わなければなりませんが)」

ということで，作者の林さんから説明をもらったところで，プログラムのコンパイル方法です。このプログラムはC言語で書かれていますが，ライブラリにBASICライブラリとDOSライブラリを使用しているので，エディタでリストを打ち込んだあとで，XCを使って以下のようにコンパイルしてください。

```
A>CC /W /Y REFONT.C
```

では使い方を説明しましょう。

まず，コマンドラインから，REFONT.Xに入力ファイル名と出力ファイル名をオプ

ションで指定して使ってください。たとえば，MINCHO.VF1フォントを変換してSAMPLE.VF1にするならば，

REFONT MINCHO.VF 1 SAMPLE.VF1

としてください。フォントの変形にはかなりの時間がかかりますので，作業は時間に余裕のあるときにしましょう。

また，最初のコードが変なフォントも存在するようで，ときどき画面がグチャグチャになりますが，変換はちゃんと行われていますので気にしないでください。さて，無事できましたか？

さらにこのプログラム，コンバートの種類を変えてほかの方法で変形することもできるようになっていて，リストに注釈の形で入っています。変えるときには適当な変換関数の注釈をとって，現在の変換関数を注釈にして，コンパイルし直してみてください。

また，START CODE，END CODE，MAX LENGTHも変えることができます。それぞれ，変換開始コード，終了コード(それぞれシフトJIS)，最大線分長です。最大線分長は，線を補間するときの各線分の最大の長さです。1024以上にすることでまったく補間されなくなりますが，直線的な写像ではそれで十分かもしれません。ただし，この最大長をあまり小さくしすぎると(10とか)，データがかさみすぎて，平気で8Mバイトなんて，フォントファイルを作成してしまうこともありますから，気をつけてくださいね(しかし，今月はいじれるパラメータの多いプログラムが多いなあ)。

それから，作成したフォントをSX-WINDOWで試用する場合は，IFM.ENVを書き換えることになります。ただし，新しいフォントを試す場合は，以前に試した番号とずらさないとうまくいかないようです。気をつけてくださいね。作者の林さんはそれで，かなり悩んだのだそうです。

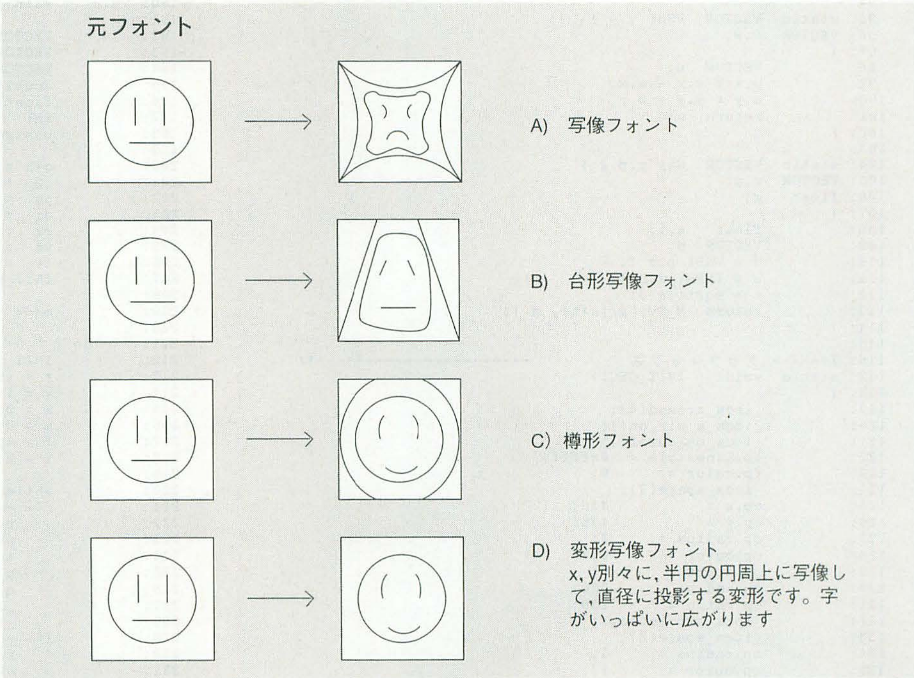
さらに，変換関数ですが，変換関数で気をつけなければならないのは，座標を(1023,1023)以内に収めなければならないということです。また，座標(1023,1023)は特別の意味をもつので，これは使用できません。実際に使用した場合に，マイナス，プラスの範囲で超えた場合は，それぞれ0，1023に変換され，さらに(1023,1023)は，(1022,1023)に強制的に変換されてしまいます。ソース内には，4つの変換関数が例として示されています(図)。

ふぅ。ちょっと説明が多くて大変だけど，これだけ説明が書かれていると改造も簡単でいいですよね。これを参考にして，フォントの角が削られた丸文字写像を作ってみるとかね。

ということで今月はここまで。さーて，映画でも観にいきましょうか，と思ったらもうとっぷり深夜でやんの。ううー，映画も観にいけない。やはりセミの霊が……(やめんか，ただでさえホラーは苦手なんじゃっ！)。

んでは，また来月。

図　フォント変形の例



リスト1　APOLLO.C

```
/*    ☆   ☆           A P O L L O          ☆   ☆

           重力場運動シミュレーション
        平成7年7月30日(日)   小枝 直隆
   ☆  ☆  ☆  ☆  ☆  ☆  ☆  ☆  ☆  ☆  ☆  ☆  */

#include        <math.h>
#include        <stdio.h>
#include        <conio.h>
#include        <iocslib.h>

/*----- 定数・変数初期化       ----------------------*/
static  struct  _lineptr        lp;
static  struct  _circleptr      cp;
static  struct  _fillptr        fp;

typedef struct {
        float   x;
        float   y;
} VECTOR;

typedef struct {
        VECTOR  z;
        VECTOR  w;
} MATRIX;

double  S_T[11] =
        {       0, 2, 4, 2,
                0, 2, 4, 2, 0, 2, 0     };

char    A_T[18] =
        {       8, 0, 4, 8, 6, 7, 5, 6, 2,
        1, 3, 2, 8, 0, 4, 8, 3, 5       };

MATRIX  ROT_T[18];

static  void    INIT_MATS( double rg )
{
        int     i,a;
        double  t,c,s;
        /*- - -   手作りｓｉｎテーブル    - - -*/
        for ( i=0; i<10; i++ ) {
                S_T[i] = sqrt(S_T[i])*0.5;
                if ( i>4 && i<8 )
                        S_T[i] *= -1;
        }
        t = rg;
        /*- - -    回転行列テーブル    - - -*/
        for ( i=0; i<18; i++ ) {
                if ( i==16 )
                        t = 5;
                a = A_T[i];
                s = t*S_T[ a ];
                c = t*S_T[a+2];
                ROT_T[i].z.x =  c;
                ROT_T[i].z.y = -s;
                ROT_T[i].w.x =  s;
                ROT_T[i].w.y =  c;
        }
        return;
}
```

```
/*----- ベクトルと行列の演算    ----------------------*/
static  float   INPR( v,w )
VECTOR  v,w;
{
        return  v.x * w.x + v.y * w.y;
}

static  VECTOR  M_MV( MATRIX m,VECTOR v )
{
        VECTOR  u;
        u.x = INPR( m.z, v );
        u.y = INPR( m.w, v );
        return  u;
}

static  VECTOR  M_SV( float s,VECTOR v )
{
        VECTOR  u;
        u.x = v.x * s;
        u.y = v.y * s;
        return  u;
}

static  VECTOR  VAD( v,w )
VECTOR  v,w;    .
{
        VECTOR  u;
        u.x = v.x + w.x;
        u.y = v.y + w.y;
        return  u;
}

static  VECTOR  VSB( v,w )
VECTOR  v,w;
{
        VECTOR  u;
        u.x = v.x - w.x;
        u.y = v.y - w.y;
        return  u;
}

static  VECTOR  GR( r,p,g )
VECTOR  r,p;
float   g;
{
        float   s,t;
        VECTOR  d;
        d = VSB( p,r );
        s = INPR( d,d );
        t = sqrt( s );
        return  M_SV( g/(s*t), d );
}

/*----- グラフィックス            ----------------------*/
static  void    INIT_GSC()
{
        _iocs_crtmod(4);
        _iocs_g_clr_on();
        _iocs_os_curof();
        lp.linestyle =  0xffff;
        fp.color =      0;
        _iocs_apage(3);
        cp.x =          456;
        cp.y =          456;
        cp.radius =     7;
        cp.color =      3;
        cp.start =      0;
        cp.end =        360;
        cp.ratio =      256;
        _iocs_circle(&cp);
        _iocs_apage(2);
        cp.radius =     3;
        cp.color =      13;
        _iocs_circle(&cp);
        _iocs_apage(0);
        _iocs_vpage(0xf);
        printf("☆  A P O L L O  ☆¥n¥n");
        return;
}

static  void    DRAW_R( r,e,c )
VECTOR  r,e;
unsigned short  c;
{
        VECTOR  p0,p1,p2;
        p0 = VAD( r,M_SV(       5   ,e ) );
        p1 = VAD( r,M_MV( ROT_T[16],e ) );
        p2 = VAD( r,M_MV( ROT_T[17],e ) );
        lp.x1 =         p0.x;
        lp.y1 =         p0.y;
        lp.x2 =         p1.x;
        lp.y2 =         p1.y;
        lp.color =       c  ;
        _iocs_line(&lp);
        lp.x1 =         p2.x;
        lp.y1 =         p2.y;
        _iocs_line(&lp);
        lp.x2 =         p0.x;
        lp.y2 =         p0.y;
        _iocs_line(&lp);
        return;
}

static  void    CLR_R( r )
VECTOR  r;
{
        fp.x1 = r.x - 5;
        fp.y1 = r.y - 5;
        fp.x2 = r.x + 5;
        fp.y2 = r.y + 5;
        _iocs_fill(&fp);
        return;
}

/*----- ジョイスティック入力   ----------------------*/
static  int     JOY_SS()
{
        int     c;
        c = (_iocs_joyget(0) ^ 0xff);
        if ( (c & 12) == 12 )
                c = c ^ 0x8c;                   /*   start   */
        if ( (c & 3) ==  3 )
                c = c ^ 0x13;                   /*   select  */
        return  c;
}

/*===== メインルーチン  ======================*/
void    main()
{
        VECTOR  rr, ev, ps, iv = { 0.7,  0 };
        VECTOR  p, q, r, v;
        VECTOR  ir = { 456,280 } ,  o = { 256,256 };
        double  rg, c2,             clk = 10; /*<- 動作周波数 */
        float   pg, qg, va, a;
        int     f,  i,  j,  k,  s = 0,  md;
        unsigned short  c;

        clk =  10 / clk;
        c2  = clk * clk;
        pg  =       15 * c2 ;
        qg  =        5 * c2 ;
        rg  =     0.01 * c2 ;
        va  = -0.0005 * clk;
        iv  = M_SV( clk,iv );
        INIT_MATS(rg);

        md = _iocs_crtmod( -1 );

    do {
        INIT_GSC();
        r = ir;
        v = iv;
        k = 0;
        a = 0;
        f = 0;
        i = 0;

        while ( (k & 0xf0) == 0 ) {
        /*- - -       星の運行        - - -*/
            ps.x = 200 * cos( a );
            ps.y = 200 * sin( a );
            a += va;
            p = VAD( o,ps );
            q = VSB( o,ps );
            _iocs_home( 8, 456-p.x, 456-p.y );
            _iocs_home( 4, 456-q.x, 456-q.y );
        /*- - -       船の運動        - - -*/
            rr = r;
            ev = M_SV( 1/sqrt(INPR( v,v )), v );
            v = VAD( v,GR( r,p,pg ) );
            v = VAD( v,GR( r,q,qg ) );
            k = JOY_SS();
            j = k & 0x0f;
            if ( j ) {
                f++;
                c = 5;
                v = VAD( v,M_MV(ROT_T[j],ev) );
            } else    c = 11;
            r = VAD( r,v );
            DRAW_R( r,ev,c );
            s = s ^ 1;
            _iocs_apage( s );
            CLR_R( rr );
            _iocs_b_up_s();
            printf("time:%6i  fuel:%5i¥n", ++i, f);
        }
    } while ( (k & 0x30) == 0 );
    _iocs_crtmod( md );
    while ( kbhit() ) getch();
    _iocs_os_curon();
}
```

リスト2 MARUKO.S

```
*---------------------------------------------------------
*     B A S I C   F N C .
* テキスト読み込み(プリンタ(24dots)ビットイメージ型)
*                 &  配列データ転送・切り出し
*                 &  漢字/TAB入り文字列用 LEFT$
*
*                         H7 / 7 / 1
*                                         小枝  直隆
*---------------------------------------------------------

                .include        iocscall.mac
                .include        doscall.mac
                .include        fdef.h

```

▶やった，新機種が出たぁ～！　あ，ちょっと気が早かったかな？
伊藤　真(21)東京都

```
        .text
        .even
* infomation table
        .dc.l   return
        .dc.l   return
        .dc.l   return
        .dc.l   return
        .dc.l   return
        .dc.l   return
        .dc.l   return
        .dc.l   return
        .dc.l   txrd_token
        .dc.l   txrd_param
        .dc.l   txrd_exec
        .dc.l   0,0,0,0,0
txrd_token:
        .dc.b   'txread',0
        .dc.b   'tpdims',0
        .dc.b   'ktleft$',0
        .dc.b   0
        .even
txrd_param:
        .dc.l   txread_par
        .dc.l   tpdims_par
        .dc.l   ktleft_par
txread_par:
        .dc.w   int_val             * X 座標
        .dc.w   int_val             * Y 座標
        .dc.w   int_val             * DX
        .dc.w   aryl_c              * データ格納領域
        .dc.w   int_ret             * 戻り値　実質格納量
tpdims_par:
        .dc.w   aryl_fic            * 転送元データ
        .dc.w   int_val             * スタート位置（０～　）
        .dc.w   int_val             * 個数（０～　）
        .dc.w   aryl_fic            * 転送先
        .dc.w   int_val             * 転送位置（０～　）
                                    * 位置・個数はすべてバイト単位
        .dc.w   int_ret             * 戻り値　実質転送量
ktleft_par:
        .dc.w   str_val
        .dc.w   int_val             * 文字数
        .dc.w   int_omt             * TAB_shift（省略可）
        .dc.w   str_ret
txrd_exec:
        .dc.l   txread
        .dc.l   tpdims
        .dc.l   ktleft
*---    t x r e a d ( ) 関数
txread:
        bsr     super
        move.l  12(sp),d0       * x1
        move.l  22(sp),d1       * y1
        move.l  32(sp),d2       * dx
        andi.l  #$07fe,d2
                    * 列数を 2046 以下の偶数に切り下げ
        movea.l 42(sp),a2
        moveq.l #0,d3
        move.w  8(a2),d3
        addq.w  #1,d3           * d3 <- dim の要素バイト数
        adda.l  #10,a2          * a2 <- dim pointer
        bsr     check_xya
        bcs     txread_err_rts  * エラー終了
        move.l  d2,ret_l
        lea.l   ret_p(pc),a0    * 正常終了設定 1
        cmpi.l  #0,d2
        beq     txread_rts      * dx = 0 の時すぐ終了
        move.l  d0,d3           * x1 を退避
        lea.l   $e00000,a1
        lsr.w   #3,d0
        lsl.l   #7,d1
        adda.w  d0,a1
        adda.l  d1,a1           * VRAM ADRS を a1 に
        move.l  d3,d0           * x1 復帰
        andi.w  #$0007,d0
        subq.w  #7,d0
        neg.w   d0              * VRAM ADRS' を d0 に
        move.l  #128,d5         * VRAM DELTA
        move.w  $e8002a,accs_mode       * アクセスモード保存
        move.w  #$ff,$e8002a            * 同時アクセスＯＦＦ
LARGE_LOOP:
        * COUNTER は d2（ただし、1ループ2カウント減）
        movea.l a1,a3           * 退避
        moveq.l #0,d3           * ワークデータ（左）
        moveq.l #0,d4           * ワークデータ（右）
        moveq.l #24,d6          * LOOP COUNTER
LOOPLF:
        lsl.l   #1,d3
        move.b  (a1),d1
        btst.l  d0,d1
        beq     JMPLF
        addq.l  #1,d3
JMPLF:
        subq.b  #1,d0
LOOPRT:
        lsl.l   #1,d4
        adda.l  d5,a1
        btst.l  d0,d1
        beq     JMPRT
        addq.l  #1,d4
JMPRT:
        addq.b  #1,d0
        subq.b  #1,d6
        bne     LOOPLF
        * d3,d4 それぞれ下位24bit に L,R データ
        ror.l   #8,d3
        move.l  d3,d1
        or.l    d4,d1
        swap.w  d1
        * d3.w, d1.w, d4.w の順でならぶデータ
        move.w  d3,(a2)+
        move.w  d1,(a2)+
        move.w  d4,(a2)+                * 配列エリアに転送
        movea.l a3,a1                   * 復帰
        subq.b  #2,d0                   *----------------------
        bcc     JMPLLP                  * VRAM ADRS
        addq.l  #1,a1                   *       /   ADRS' 操作
        moveq.l #7,d0                   *----------------------
JMPLLP:
        subq.w  #2,d2
        bne     LARGE_LOOP
        move.w  accs_mode(pc),$e8002a   * アクセスモード復帰
txread_rts:
        bsr     user
        moveq.l #0,d0                       * 正常終了設定 2
return:
        rts
*---    t p d i m s ( ) 関数
tpdims:
        movea.l 12(sp),a0
        bsr     dim_ptr_sz
        movea.l a0,a1       * a1 <- s_dim pointer
                            *       ( s_dim = ソース配列 )
        move.l  d2,d1       * d1 <- s_dim の要素バイト数
        movea.l 42(sp),a0
        bsr     dim_ptr_sz
        movea.l a0,a2               * a2 <- d_dim pointer
                            * ( d_dim = ディスティネーション配列 )
*       move.l  d2,d2               * d2 <- d_dim の要素バイト数
        move.l  22(sp),d3           * s_dim オフセット
        move.l  32(sp),d4           * 転送バイト数
        move.l  52(sp),d5           * d_dim オフセット
        bsr     check_areas
        bcs     tpdims_err_rts      * エラー終了
        move.l  d4,ret_l
        lea.l   ret_p(pc),a0
        moveq.l #0,d0               * 正常終了設定
        adda.l  d3,a1               * s_pointer オフセット
        adda.l  d5,a2               * d_pointer オフセット
        cmp.l   d5,d3
        bcc     TRANS_FF_ST         * d3 >= d5 なら前から転送
                                    *       そうでない時は後ろから
        adda.l  d4,a1
        adda.l  d4,a2
        bra     TRANS_RW_ST
LOOP_RW:
        move.b  -(a1),-(a2)
TRANS_RW_ST:
        subq.l  #1,d4
        bcc     LOOP_RW
        rts
LOOP_FF:
        move.b  (a1)+,(a2)+
TRANS_FF_ST:
        subq.l  #1,d4
        bcc     LOOP_FF
        rts
        *--- tpdims サブ ---
dim_ptr_sz:
        moveq.l #0,d0
        move.w  6(a0),d0
        moveq.l #0,d2
        move.w  8(a0),d2
        addq.w  #1,d2
        adda.l  #10,a0              * a0 <- dim pointer
        subq.b  #1,d0
        beq     LAB_dim
        add.l   d2,d2
        add.l   d2,d2
        subq.b  #3,d0
        beq     LAB_dim
        add.l   d2,d2               * d2 <- dim の要素バイト数
LAB_dim:
        rts
*---    k t l e f t $ ( ) 関数
```

▶ゲッ！　大家から電話賃として８万円も請求されてしまった……。のちに機械の故障だったことがわかったのだが，今月は３万円，いいかげん直せ！　　和田　智(20)長野県

```
ktleft:
        move.l  32(sp),d3
        cmpi.w  #1,26(sp)
        beq     TAB_shift
        moveq.l #0,d3
TAB_shift:
        movea.l 12(sp),a1
        move.l  22(sp),d1
        lea.l   ret_str_area(pc),a2
        move.l  a2,ret_l

        cmpi.l  #256,d1
        bcs     LEN_ok
        moveq.l #255,d1
LEN_ok:
        cmpi.b  #0,d1
        beq     END_code_set

LOOP_ktl:
        move.b  (a1)+,d2
        cmpi.b  #$09,d2
        beq     TAB_process

        cmpi.b  #$80,d2
        bcs     NO_kanji
        cmpi.b  #$a0,d2
        bcs     YES_kanji
        cmpi.b  #$e0,d2
        bcs     NO_kanji

YES_kanji:
        addq.b  #1,d3
        subq.b  #1,d1
        beq     END_code_set
        move.b  d2,(a2)+
        move.b  (a1)+,(a2)+
        bra     JMP_ktl

NO_kanji:
        move.b  d2,(a2)+
JMP_ktl:
        addq.b  #1,d3
        subq.b  #1,d1
        bne     LOOP_ktl
END_code_set:
        move.b  #0,(a2)

        lea.l   ret_p(pc),a0
        moveq.l #0,d0
        rts

TAB_process:
        move.b  d2,(a2)+
        moveq.l #8,d2
        andi.l  #7,d3
        sub.b   d3,d2
        moveq.l #0,d3
        sub.b   d2,d1
        bhi     LOOP_ktl

        bra     END_code_set

        *=== エラー終了処理 ===
txread_err_rts:
        bsr     user
tpdims_err_rts:
        lea.l   ret_p(pc),a0
        moveq.l #1,d0
        rts

        *=== ユーザ→スーパーバイザ ===
super:
        clr.l   -(sp)
        DOS     _SUPER
        addq.l  #4,sp
        move.l  d0,sspsave
        rts

        *=== スーパーバイザ→ユーザ ===
user:
        move.l  sspsave(pc),-(sp)
        DOS     _SUPER
        addq.l  #4,sp
        rts

        *=== txread エラーチェック ===
check_xya:
        cmpi.w  #0,d0
        blt     zahyou_err
        cmpi.w  #1000,d0
        bgt     zahyou_err
        cmpi.w  #0,d1
        blt     zahyou_err
        cmpi.w  #1000,d1
        bgt     zahyou_err

        moveq.l #$01,d4
        and.w   d0,d4                   * x1 が偶数でない時エラー
        bne     zahyou_err

        move.l  d2,d4
        add.l   d4,d4
        add.l   d2,d4                   * d4 <- dx*3 (= d2*3)
        cmp.l   d3,d4
        bhi     data_taran_err          * データ領域不足エラー

        andi.b  #$fe,ccr
        rts

        *=== tpdims エラーチェック ===
check_areas:
        cmpi.l  #0,d3
        blt     areas_err
        cmpi.l  #0,d4
        blt     areas_err
        cmpi.l  #0,d5
        blt     areas_err
        move.l  d3,d0
        add.l   d4,d0
        cmp.l   d1,d0                   * s_dimの領域オーバー?
        bhi     areas_err
        move.l  d5,d0
        add.l   d4,d0
        cmp.l   d2,d0                   * d_dimの領域オーバー?
        bhi     areas_err

        andi.b  #$fe,ccr
        rts

        *--- エラー処理サブ ---
zahyou_err:
        lea.l   errmes1(pc),a1
        bra     err_process

data_taran_err:
        lea.l   errmes2(pc),a1
        bra     err_process

areas_err:
        lea.l   errmes3(pc),a1

err_process:
        move.l  #-1,ret_l
        ori.b   #1,ccr
        rts


        .data

        *=== ワークエリア(変数) ===
ret_str_area:
        .ds.b   256
sspsave:
        .ds.l   1
accs_mode:
        .ds.w   1
ret_p:
        .dc.w   0
ret_h:
        .dc.l   0
ret_l:
        .dc.l   0

        *=== ワークエリア(定数) ===
errmes1:
        .dc.b   '座標値が範囲外です',0,0
errmes2:
        .dc.b   '配列領域が不足しています',0,0
errmes3:
        .dc.b   '配列領域の指定値が異常です',0,0


        .end
```

リスト3 まるコ.BAS

```
1000 /*--------------------------------------------------------
1010 /*テキストファイルまるまる(縮小)ハードコピープログラム
1020 /*     「 ま る コ 」
1030 /*                平成7年7月30日(日)  小技 直隆
1040 /*--------------------------------------------------------
1050 ANY_INIT()
1060 int wd,wdd,dt,dtt,bp=0,lpt,flp,fmt=0
1070 int mm,dd,mc,nm8
1080 int i,j,k
1090 str nh,s[255],cr
1100 cr=chr$(10)+chr$(13)
1110 str border[255]
1120 dim char acc(4208)          :/*   1400*3+8+1  =  4209 > 700*3 =
2100
1130 dim float buf(41999)        :/*   1400*3*80/8 = 42000
1140 dim char pcon(16)={27,62,27,37,57,16,27,92, 0, 0,27,74, 0, 0}
1150 int ps,ww,gs,st=1,ed=0
1160 int nm,nf,ns
1170 int ls=0,moe=2,pch
1180 int pz,sh,sf,poe
1190 str dvi[255]="h:¥TXH_DVI¥"  :/*   事前にサブディレクトリを作っ
てね
1200 str fnm[256]
1210 repeat
1220   input "データファイル名(フルパスで):",fnm
1230   flp=DF_OPEN()
1240 until flp>=0
1250 /*---  M a i n  L o o p  --------------------
1260 repeat
1270   cls
1280   print "= = =  テキストファイルまるまる(縮小)ハードコピ
ープログラム  = = =";cr
1290   print "データファイル:";fnm
1300   fmt=FORMAT_MENU(fmt)
1310   DATA_OUT()
1320   print cr;"処理がすべて終了しました。プログラムを終了しますか?
```


▶後輩がX68030を買いました。DOS/Vを買えとみんなにいわれていたのに……。これでシャープから新型が出たら，どうするんだろうと思ってしまいました。
上岡 幸二(28)神奈川県

```
(y/n)"
1330 until YNKEY_IN()=-1
1340 end
1350 /*========================================
1360 func int BUF_TENSO(i;int)
1370 if (i<3)+mc then return(i)
1380 int j
1390 j=(csrlin-i) shl 4
1400 txread(nm8,j,dt,acc)
1410 bp=bp+tpdims(acc,0,dtt,buf,bp)
1420 txread(nm8,j+24,dt,acc)
1430 bp=bp+tpdims(acc,0,dtt,buf,bp)
1440 return(i-3)
1450 endfunc
1460 /*----------------------
1470 func int MJKEY_IN()
1480 int m,n,o,p
1490 msstat(m,n,o,p)
1500 return(strig(1)+strig(2)-o-p)
1510 endfunc
1520 /*----------------------
1530 func str TOKCODE(s;str,c;int)
1540 int i,j,k
1550 str t[255]=""
1560 j=len(s)
1570 for i=1 to j
1580   k=asc(mid$(s,i,1))+c
1590   t=t+chr$(k shr 8)+chr$(k and 255)
1600 next
1610 return(t)
1620 endfunc
1630 /*----------------------
1640 func DATA_OUT()
1650 int i=0,j,k,l=0,m,n,p=1,q
1660 str s[255],t[255],u,n0,n1
1670 n1=space$(nm shl 2)
1680 m=(gs*3 shr 1)-2-sh-sf
1690 fseek(flp,0,0)
1700 mm=1:mc=M_CHK()
1710 for n=1 to 9999
1720   if freads(s,flp)=-1 then break
1730   j=1
1740   if n mod ns then n0=chr$(9) else {
1750     u=TOKCODE(str$(n),nf)
1760     n0=space$(12-len(u) shr 1)+u+nh
1770   }
1780   n0=left$(n0,nm8)
1790   repeat
1800     t=ktleft$(s,dd):s=mid$(s,len(t)+1,255)
1810     if l=0 then i=i+HEAD_OUT(p)
1820     if j then print n0;t:j=0 else print n1;t
1830     i=BUF_TENSO(i+1)
1840     l=l+1
1850     if l=m then {
1860       i=BUF_TENSO(i+FOOT_OUT(p))
1870       if p mod ps=0 then {
1880         if LPT_OUT() then {
1890           print "以降の処理を続行しますか？ (y/n)";cr
1900           q=YNKEY_IN()
1910         } else q=feof(flp)*2
1920         if q=-2 then n=9999:break
1930         mm=mm+1
1940         mc=M_CHK()
1950       }
1960       p=p+1:l=0:i=0
1970     }
1980   until s=""
1990   while MJKEY_IN()
2000   endwhile
2010 next
2020 if q=-2 then return()
2030 /*- - 残りの空白 - - - -
2040 if l then {
2050   while l<m
2060   print
2070     l=l+1
2080     i=BUF_TENSO(i+1)
2081     while MJKEY_IN()
2082     endwhile
2090   endwhile
2100   i=BUF_TENSO(i+FOOT_OUT(p))
2110 }
2120 print cr;cr
2130 while p mod ps>0
2140   j=gs shr 1
2150   for i=1 to j
2160     BUF_TENSO(3)
2170   next
2180   p=p+1
2190 endwhile
2200 LPT_OUT()
2210 endfunc
2220 /*----------------------
2230 func int LPT_OUT()
2240 print border
2250 if mc then return(0)
2260 int a,b,c,d,i,l,m,r=0
2270 str j[255],k[255]
2280 j=TOKCODE(str$(mm),&H821F)
2290 j=j+"枚目"
2300 bp=0
2310 if ls then {
2320   lpt=fopen(dvi+"D_"+str$(mm)+".dvi","c")
2330 } else {
2340   lpt=fopen("lpt","w")
2350 }
2360 fwrite(pcon,6,lpt)
2370 a=dt*ps:b=a shr 8:c=a and 255
2380 tpdims(pcon,6,8,acc,0)
2390 acc(2)=pch*((mm+poe) and 1):acc(6)=b:acc(7)=c
2400 d=a*3+9
2410 acc(d-1)=10
2420 c=dtt*gs
2430 l=0
2440 print cr,"－－  ";
2450 print j;mid$("の ＤＶＩファイルを作成を 印字",24-ls*23,23);"中
－－"
2460 print ,"マウス／ジョイスティクのボタンを押すと、中断します。";cr
2470 fputc(10,lpt)
2480 repeat
2490   b=l*dtt
2500   for i=0 to ps-1
2510     tpdims(buf,i*c+b,dtt,acc,8+i*dtt)
2520   next
2530   fwrite(acc,d,lpt)
2540   m=MJKEY_IN()
2550   if m then {
2560     print "！！  ボタンが押されました。";j;
2570     print "の処理を中止しますか？（y/n）";
2580     r=YNKEY_IN()
2590     r=r and 1
2600     print chr$(13);chr$(5);
2610   }
2620 l=l+1
2630 until (l=gs)-r
2640 if r then {
2650   print ,"▽  ▽  ▽",j;"の処理を中止      ▽  ▽  ▽";cr
2660 } else{
2670   print ,"＝＝  ";j;
2680   print mid$("ファイル 作成 印字",14-ls*13,13);"終了  ＝＝";cr
2690 }
2700 fputc(12,lpt)
2710 fclose(lpt)
2720 return(r)
2730 endfunc
2740 /*----------------------
2750 func int DF_OPEN()
2760 int f
2770 f=fopen(fnm,"r")
2780 if f<0 then {
2790   print cr;"エラー：ファイル（ ";fnm;" ）がみつかりません";cr;cr
2800 }
2810 return(f)
2820 endfunc
2830 /*----------------------
2840 func int HEAD_OUT(n;int)
2850 print border
2860 if sh then print P_ZUME((TOKCODE(str$(n),&H821F)));cr
2870 print
2880 return(sh+1)
2890 endfunc
2900 /*----------------------
2910 func int FOOT_OUT(n;int)
2920 print
2930 if sf then print cr;P_ZUME((TOKCODE(str$(n),&H821F)))
2940 return(sf+1)
2950 endfunc
2960 /*----------------------
2970 func str P_ZUME(s;str)
2980 return(space$(((wd-len(s)) shr 1)*pz+nm)+s)
2990 endfunc
3000 /*----------------------
3010 func int M_CHK()
3020 return((mm<st)+(mm>ed)+((mm and 1)=moe))
3030 endfunc
3040 /*----------------------
3050 func int YNKEY_IN()
3060 str k
3070 int r
3080 KB_CLR()
3090 repeat
3100   k=inkey$
3110   r=(k="y")+(k="Y")+((k="n")+(k="N"))*2
3120 until r
3130 return(r)
3140 endfunc
3150 /*----------------------
3160 func KB_CLR()
3170 repeat
3180 until inkey$(0)=""
3190 endfunc
3200 /*----------------------
3210 func ANY_INIT()
3220 width 96
3230 console ,,0
3240 mouse(4)
3250 error off
3260 endfunc
3270 /*----------------------
3280 func int FORMAT_MENU(n;int)
3290 int i,m,dm,s=8
3300 str k
3310 dim str msg(7)[255]={
3320   "Ａ４・２ページ／枚・行番号あり・ルーズリーフ",
3330   "Ａ４・２ページ／枚・行番号あり",
3340   "Ａ４・２ページ／枚・行番号なし・ルーズリーフ",
3350   "Ａ４・３ページ／枚・いっぱい＾２",
3360   "Ａ４・半角６４文字／行（２ページ／枚）",
3370   "Ｂ５・半角６４文字／行（２ページ／枚）",
3380   "Ｂ５・２ページ／枚・行番号あり・ルーズリーフ",
3390   "Ｂ５・２ページ／枚・いっぱい＾２"
3400 }
3410 for i=0 to s-1
3420   locate 5,i*2+5,0:color 3-(i=n)*12:print msg(i)
3430 next
3440 m=n
3450 KB_CLR()
3460 repeat
3470   k=inkey$
3480   dm=(k=chr$(30))+(k="8")-(k=" ")-(k=chr$(31))-(k="2")
3490   if dm then {
3500     locate 5,m*2+5:color 3:print msg(m)
3510     m=(m+dm+s) mod s
3520     locate 5,m*2+5:color 15:print msg(m)
3530   }
3540 until k=chr$(13)
3550 locate ,,1:color 3
3560 ps=2:ww=1400:gs=80
3570 nm=2:nf=&HF200:ns=5
3580 pch=48:pz=1:sh=0:sf=2:poe=0
3590 wdd=ww-pch:wd=(wdd shr 3)¥ps
3600 switch m
3610   case 0:break
3620   case 1:pch=0:wdd=ww-pch:wd=(wdd shr 3)¥ps:break
3630   case 2:nm=0:wdd=ww-pch:wd=(wdd shr 3)¥ps:break
3640   case 3:ps=3:nm=0:pch=0:wdd=ww-pch:wd=(wdd shr 3)¥ps:break
```

▶CD-ROMドライブを購入したばかりだというのに「Mach-２」がほしくなった。MDを買っていなければよかったのに……(いちばん使っていない)。　三宅　洋史(22)福岡県

```
 3650   case 4:wd=64+6:ns=1:nf=&H8000:wdd=ww-pch:ps=(wdd shr 3)¥wd:bre
ak
 3660   case 5:ww=1208:gs=68:wd=64+6:ns=1:nf=&H8000:wdd=ww-pch:ps=(wdd
 shr 3)¥wd:break
 3670   case 6:ww=1208:gs=68:wdd=ww-pch:wd=(wdd shr 3)¥ps:break
 3680   case 7:ww=1208:gs=68:nm=0:pch=0:wdd=ww-pch:wd=(wdd shr 3)¥ps:b
reak
 3690 endswitch
 3700 /*- - - - - - - - - - -
 3710 locate 0,25
 3720 input "スタート枚目：",st
 3730 input "エンド枚目(スタート<エンド とするとプレビューアになりま
す）：",ed
 3740 input "奇数枚目だけ...0  偶数枚目だけ...1  全部の枚目...それ以外
：",moe
 3750 input "プリンタに直接...0  一旦ファイルにする...1 ：",ls
 3760 dt=wdd¥ps:dtt=dt*3:dd=wd-nm*3
 3770 nm8=nm shl 3
 3780 nh=TOKCODE(":",nf)+" "
 3790 border=left$("  ",nm)+"|"+string$(wd-2,"-")+"|"
 3800 return(m)
 3810 endfunc
```

リスト3 REFONT.C

```
/*
 * 写像フォントコンバータ refont.x
 */
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <graph.h>
#include <doslib.h>

/*
 * 定数定義
 */
#define PI 3.14159265

#define FONT_NUM 4418
#define START_CODE 0x0000
#define END_CODE   0xFFFF
#define MAX_LENGTH 100

/*
 * グローバル変数定義
 */
/* 入出力フォントテーブル */
struct {
  unsigned short code;
  unsigned long i_offset;
  unsigned long o_offset;
} font_table[FONT_NUM];
/* 入出力ファイル名 */
char ifn[256], ofn[256];
/* 入出力ファイルポインタ */
FILE *ifp, *ofp;
/* 書き込みオフセット */
unsigned long offset = 0;

/*
 * 関数プロトタイプ宣言
 */
void init_table( void );
void change_font( int );
void line2( int, int, int, int );
void get_point( int, int *, int * );
void put_point( int, int, int );
void change_point( int *, int * );
void change_func( int *, int * );
unsigned short get_ushort( void );
unsigned long get_ulong( void );
unsigned short put_ushort( unsigned short );
unsigned long put_ulong( unsigned long );
void uses( void );
void error( int );

/*
 * メイン関数
 */
void main(int argc, char **argv)
{
  int i;

  /* コマンドライン取り出し */
  if(argc != 3)
    uses();
  strcpy(ifn, argv[1]);
  strcpy(ofn, argv[2]);
  /* ファイルのオープン */
  if((ifp = fopen(ifn, "rb")) == NULL)
    error(0);
  if((ofp = fopen(ofn, "wb")) == NULL)
    error(1);
  /* フォントテーブル初期化 */
  init_table();

  /* 処理開始 */
  C_WIDTH(1);
  offset = 0;
  for(i = 0; i < FONT_NUM; ++i) {
    if(START_CODE > font_table[i].code)
      continue;
    if(END_CODE < font_table[i].code)
      continue;
    printf("SHIFT JIS CODE:%4X", font_table[i].code);
    if(font_table[i].i_offset == 0xFFFFFFFF)
      printf("...NO DATA¥n");
    else {
      font_table[i].o_offset = offset;
      change_font(i);
      printf("...DONE¥n");
    }
  }
  /* テーブル書き込み */
  if(fseek(ofp, 2, SEEK_SET) != 0)
    error(2);
  for(i = 0; i < FONT_NUM; ++i)
    put_ulong(font_table[i].o_offset);

  fcloseall();
}

/*
 * フォントテーブル初期化
 */
void init_table( void )
{
  unsigned short mode;
  int code;
  int i;

  /* 水準読み込み */
  mode = get_ushort();
  switch(mode) {
  case 1:
    code = 0x8140;
    break;
  case 2:
    code = 0x989F;
    break;
  default:
    error(5);
  }
  put_ushort(mode);
  printf("第%d水準フォントです。¥n", mode);

  for(i = 0; i < FONT_NUM; ++i) {
    font_table[i].code = code;

    switch(code & 0xFF) {
    case 0x7E:
      code += 2;
      break;
    case 0xFC:
      if(code == 0x9FFC)
        code = 0xE040;
      else
        code += 0x140 - 0xFC;
      break;
    default:
      ++code;
      break;
    }

    font_table[i].i_offset = get_ulong();
    put_ulong(font_table[i].o_offset = 0xFFFFFFFF);
  }
}

/*
 * フォント変換処理
 */
void change_font(int n)
{
  int x0, y0, x1, y1;
  int x, y;

  if(fseek(ifp, (long)font_table[n].i_offset + 0x450A, SEEK_SET) != 0)
    error(2);
  /* 読みだし、書き込みバッファのクリア */
  get_point(0, &x, &y);
  put_point(0, x, y);
  wipe();

  while(1) {
    get_point(1, &x0, &y0);
    if((x0 == 1023) && (y0 == 1023))
      break;
    x1 = x0;
    y1 = y0;
    while(1) {
      get_point(1, &x, &y);
      if((x == 1023) && (y == 1023))
        break;
      line2(x1, y1, x, y);
      x1 = x;
      y1 = y;
    }
    line2(x1, y1, x0, y0);
    put_point(1, 1023, 1023);
  }
  put_point(1, 1023, 1023);
  put_point(2, x, y);
}

/*
 * ラインの点補間
 */
void line2(int x1, int y1, int x2, int y2)
{
  int dx, dy, x, y, x0, y0;
  double l;
  int i, n;

  dx = x2 - x1;
  dy = y2 - y1;
  l = sqrt((double)(dx * dx) + (dy * dy));
  n = l / MAX_LENGTH;
  if(l - n * MAX_LENGTH != 0.)
    ++n;
  x0 = x1;
```

▶どうして就職活動でX68000について力説せにゃならんのか。「所有パソコンを教えてください」「X68000です」「ああ，あのゲーム用の……」「(怒)」。そしてここからX68000の素晴らしさを説くことになる(どこに行っても)。　神谷　正樹(21)愛知県

```
  y0 = y1;
  change_point(&x0, &y0);
  for(i = 0; i < n; ++i) {
    x = x1 + ((double)dx * i / n) + 0.5;
    y = y1 + ((double)dy * i / n) + 0.5;
    change_point(&x, &y);
    put_point(1, x, y);
    line(256 + x0/2, y0/2, 256 + x/2, y/2, 15, 0xffff);
    x0 = x;
    y0 = y;
  }
  change_point(&x2, &y2);
  line(256 + x0/2, y0/2, 256 + x2/2, y2/2, 15, 0xffff);
}

/*
 * 座標入出力
 */
void get_point(int mode, int *x, int *y)
{
  static int phase = 0;
  static unsigned long data;

  if(mode == 0) {
    phase = 0;
    return;
  }
  switch(phase) {
  case 0:
    phase = 1;
    data = (unsigned long)get_ushort() << 16;
    data |= get_ushort();
    *x = (data >> 22) & 0x3FF;
    *y = (data >> 12) & 0x3FF;
    break;
  case 1:
    phase = 2;
    data = (data << 16) | get_ushort();
    *x = (data >> 18) & 0x3FF;
    *y = (data >> 8) & 0x3FF;
    break;
  case 2:
    phase = 3;
    data = (data << 16) | get_ushort();
    *x = (data >> 14) & 0x3FF;
    *y = (data >> 4) & 0x3FF;
    break;
  case 3:
    phase = 0;
    data = (data << 16) | get_ushort();
    *x = (data >> 10) & 0x3FF;
    *y = data & 0x3FF;
    break;
  }
}
void put_point(int mode, int x, int y)
{
  static int phase = 0;
  static unsigned short data;
  unsigned long point;

  if(mode == 0) {
    phase = 0;
    return;
  } else if(mode == 2) {
    if(phase != 0)
      put_ushort(data);
    phase = 0;
    return;
  }

  x &= 0x3FF;
  y &= 0x3FF;
  point = (x << 10) | y;
  switch(phase) {
  case 0:
    phase = 1;
    put_ushort((unsigned short)(point >> 4));
    data = point << 12;
    break;
  case 1:
    phase = 2;
    data |= (point >> 8) & 0x0FFF;
    put_ushort(data);
    data = point << 8;
    break;
  case 2:
    phase = 3;
    data |= (point >> 12) & 0x00FF;
    put_ushort(data);
    data = point << 4;
    break;
  case 3:
    phase = 0;
    data |= (point >> 16) & 0x000F;
    put_ushort(data);
    put_ushort((unsigned short)point);
    break;
  }
}
void change_point(int *x, int *y)
{
  change_func(x, y);
  if(*x < 0)
    *x = 0;
  else if(*x > 1023)
    *x = 1023;
  if(*y < 0)
    *y = 0;
  if(*y > 1023)
    *y = 1023;
  if((*x == 1023) && (*y == 1023))
    *x = *y = 1022;
}

/*
 * 座標変換関数
 */
/* 1 ? a,bを増やすと、縦,横のへこみ率が変化します。(0～100) */
void change_func(int *x, int *y)
{
  static a = 30, b = 30;
  int x1, y1, x2, y2;

  x1 = *x;
  y1 = *y;
  x2 = ((y1 - 512) * (y1 - 512) * a +
        512 * 512 * (100 - a)) / 512 / 512;
  y2 = ((x1 - 512) * (x1 - 512) * b +
        512 * 512 * (100 - b)) / 512 / 512;
  *x = 512 + (x1 - 512) * x2 / 100;
  *y = 512 + (y1 - 512) * y2 / 100;
}
/* 2 台形フォント aは下底に対する上底の割合(0～100)
void change_func(int *x, int *y)
{
  static a = 70;

  *x = 512 + (*x - 512) * ((100 - a) * *y + 1023 * a) / 1023 /
100;
}*/
/* 3 樽形フォント aを減らすと、横の膨らみ率が上がります。(0～)
void change_func(int *x, int *y)
{
  static a = 512;
  int x1, y1, r;

  x1 = *x - 512;
  y1 = *y - 512;
  r = 512 + a;
  *x = 512 + x1 * (sqrt((double)r * r - y1 * y1) - a) / 512;
}*/
/* 4 ?
void change_func(int *x, int *y)
{
  *x = 512 * (1 - cos(*x * PI / 1023.));
  *y = 512 * (1 - cos(*y * PI / 1023.));
}*/

/*
 * ファイル入出力
 * インテルの呪いからの解放
 */
unsigned short get_ushort()
{
  unsigned char data[2];

  if(fread(data, 1, 2, ifp) != 2)
    error(3);
  return(((unsigned short)data[1] << 8) | data[0]);
}
unsigned long get_ulong()
{
  unsigned long data;

  data = get_ushort();
  data |= (unsigned long)get_ushort() << 16;
  return(data);
}
unsigned short put_ushort(unsigned short data)
{
  unsigned char data2[2];

  data2[0] = data & 0xFF;
  data2[1] = (data >> 8) & 0xFF;
  if(fwrite(data2, 1, 2, ofp) != 2)
    error(4);
  offset += 2;
  return(data);
}
unsigned long put_ulong(unsigned long data)
{
  put_ushort((unsigned short)data);
  put_ushort((unsigned short)(data >> 16));
  return(data);
}

/*
 * 使用法表示
 */
void uses( void )
{
  printf("refont <input filename> <output filename>¥n");
  exit(1);
}

/*
 * エラー
 */
void error(int n)
{
  switch(n) {
  case 0:
    printf("入力ファイル<%s>がオープンできません。", ifn);
    break;
  case 1:
    printf("出力ファイル<%s>がオープンできません。", ofn);
    break;
  case 2:
    printf("シークエラー!!¥n");
    break;
  case 3:
    printf("読み込みエラー!!¥n");
    break;
  case 4:
    printf("書き込みエラー!!¥n");
    break;
  case 5:
    printf("ファイルフォーマットが異常です。¥n");
    break;
  }
  fcloseall();
  exit(1);
}
```

▶MOに続いてCD-ROMドライブを1万円で購入。おかげでデータはますます増える一方。FDが足りなくてしかたがなかったときのように，いまはMOが不足気味。あぁ，歴史は繰り返すのね。
津村　忠蔵(20)佐賀県

lisp一夜漬け 1

Lispプログラムの書き方

Tamura Kento 田村 健人

X68000では意外と「それほどマイナーではない」言語Lisp
使い方次第では非常に強力だが，癖の強い処理系だ
ここではdivから少し離れてLisp一般への入門から始めてみよう

ということで，Lispの連載である。div-lispに限った話をするのではなく，Lisp一般について解説する(というか，div-lispって連載を書けるほど奥深いものではない)。とりあえずLispでプログラムを書けることを目標にして，3～4回程度で終了する予定である。そもそも筆者はあまりLispに詳しくないので，あまり難しいことは書けない。難しいことは書けないが，簡単なことを難しく書いてしまうことはあるかもしれない。

対象としている読者は，なんらかのプログラミングを経験していて(関数・変数という概念がわかる)，Lispインタプリタに触れる機会を持つ人である。すでにLispでプログラムを書ける人は対象外だ。

Lispの世界

C，C++，BASIC，Pascal，COBOL，FORTRAN，Lisp，Perl，REXX。現在，代表的なプログラミング言語というと，こんなものか？*1　これらのなかで，Lispはほかの言語とはまったく異なった書き方，考え方を必要とする。あまりに特殊なため，Lisp専用のハードウェアも作られた。

FORTRANが生まれた1956年のわずか3年後，1959年にLispは生まれた。C言語が1972年であるから，わりと歴史の長い言語なのである。(ちなみに，COBOL：1959年，BASIC：1964年，Pascal：1968年，MS-BASIC：1975年，C++：1980年である)

Lispは，LISt Processorという語源が表すように，「リスト」というデータ型を扱うことに長けている。Lispのプログラム自身もリストでできているので，プログラム中でプログラムを生成して実行するという芸当が簡単に行える。人工知能の分野でよく用いられるのは，この芸のおかげである。

Lispは処理系ごとの方言が強い。一応，Common Lispという基準があるにはあるが，基準としては高すぎた。Common Lispには，たとえば分数や複素数まで含まれているのである。これは実装者にとっては酷だ。ほかのメジャーな言語に比べるとフリーソフトの処理系が多いLispであるから，実装が面倒な機能は実装されない。

＊1：筆者の偏見が大いに含まれている。

X680x0でのLisp

筆者が知っている範囲で，X680x0で利用できるLisp処理系を紹介しよう。

・AI-68K (SHARP/BUG)

唯一の市販処理系である。StaffLISPというLispインタプリタ/コンパイラと，OPS PRO-68kというエキスパートシステム構築ツールで構成される。意味もなく厚いパッケージで，驚愕する値段(定価188,000円)ではあるが，IOCSを利用できるなど，なかなか侮れない製品である。個人でこの製品を持っている人はおそらくいないだろう。

・GNU Emacs/NEmacs/Mule (F.S.F./半田剣一ら)

超多機能エディタである。Emacs-Lispのインタプリタを内蔵しており，基本的な部分以外はLispで処理を行う。F.S.F.が作成しているのがGNU Emacsで，それを日本語化したものがNEmacs，多国語化したものがMuleである。Emacs-Lisp上での違いは，日本語化・多国語化に伴う変更が主で，大きな違いはない。

Human68k上では，NEmacsを移植したもの，Muleを移植したもの，Emacsを独自にms-kanji(シフトJIS)対応にして移植したもの，の3系統がある。

起動時から存在する＊scratch＊バッファでLispプログラムを入力してC-jを押すと，それを実行してくれる。

・SECDR-scheme (Atsusi Moriwaki/ぱとらっしゅ)

schemeという言語はLispの方言のひとつである。方言とはいえけっこう違うので，Lispの学習には「？」である。

・xlisp (David Michael Betz/松原弘明)

オブジェクト指向プログラミングもできる処理系である。

・sxlisp (David Michael Betz/Niels Mayer/沖@沖)

上記xlispを，SX-WINDOW上で動くようにしたもの

である。

・QuTERM（廃人王/David Michael Betz）

SX-WINDOW上のいわゆる通信ソフトであるが，自動運転の制御用にxlispを組み込んでいる。単にLispインタプリタとしても利用できる。Lisp入門のドキュメントが同梱されている。

・Hokkaido Common Lisp（山本強/遠藤隆）

Common Lispの一種である。

・div(田村健人)

貧弱なオリジナルのLisp処理系div-lispを搭載したファイラである。あまりにも基本的な部分が欠如していたり，Lispインタプリタにインタラクティブなインタフェイスがないので，Lispの学習にはまったく適さない。新たな言語を作って載せるよりは，既存言語のサブセットのほうが知識が生かせる，ぐらいに考えてくれればうれしい。

・その他

NetBSD-X68030上なら，UNIX用に作られているLisp処理系をコンパイルするのは難しくないはずである。

なお，この記事を読むときは，div以外のLispインタプリタを用意しておいてほしい。ここで紹介したほとんどがNIFTY-Serve FSHARPRに登録されている。現在発売中の書籍「SX-WINDOW ver.3.1 開発キット」のCD-ROMにはNEmacs/Mule/sxlisp/QuTERMが収録されている。

関数という概念

プログラミング言語は「手続き型言語」「関数型言語」「論理型言語」「オブジェクト指向言語」などに分類することができる。

Lispは関数型言語に分類される。関数を組み合わせることによりプログラミングする言語なのである。

関数とはなんなのかを考えてみよう。数学の世界の関数を簡単にいうと「いくつかの値を与えると，あるひとつの値が定まるという関係」である。プログラミング言語の世界の関数は一般的に「0個以上の値を与えると，なにかをして，ひとつの値を返してくれるもの」である。与える値を「引数」「parameter」「argument」，返ってくる値を「返り値」などと呼ぶ。

「いくつかの値を持って関数の頭に飛んでいって，中身を実行して，値をひとつ持って帰ってくる」という認識は，手続き型言語の考え方である。Lispではこういう認識は捨ててほしい。数学と同じ考え方で構わない。

書き方

Lispインタプリタに，

(+ 345 654)

と入力し，最後に改行キー(Emacsの場合はC-j)を押してみてほしい。おそらく画面には999と表示されて，再び入力待ちになるだろう。

今，あなたはLispインタプリタに「S式(Symbolic-expression)」を入力した。インタプリタはS式を「評価(evaluate)」し，評価結果の999を画面に表示したのである。

LispのプログラムはS式(単に「式」ともいう)の集まりで，「プログラムを実行する」というのは入力のS式を順に評価することである。

上記の式は「数値の345と654を加算する」という意味である。

再びLispインタプリタに向かって，

345

と入力する。結果，345と表示されるはずである。これは，「345を評価すると345になる」ということを表す。単に数値だけでもS式であり，評価結果はそれ自身となる。

ともかく，

(関数名 引数1 引数2 ……)

1) 全体を括弧で囲む

2) 関数名，引数どうしは空白で区切る

と書けば関数を呼び出すことができるのである。引数の数は関数による。加算を行う関数が'+'であり，引数は可変長である。

(+ 345 654 1 100 11)

というように3個以上の引数を与えることもできる。

関数の引数も関数呼び出しでよい。関数を呼び出すときはまず引数が評価され，その結果が関数に渡される。関数の引数はすべて評価される。つまり，「12×80+20/5」を計算したいときには，

(+ (* 12 80) (/ 20 5))

と書ける，ということである。乗算をする関数'*'，除算をする関数'/'の評価結果をそのまま加算の引数にできる。C言語やBASICなどに比べるとずいぶん冗長な書き方になるが，これは我慢してもらいたい。これがLispのポリシーである。演算の優先順位を考慮する必要がないという利点もある。

変数

関数内で変数を宣言して値を代入するときは，

(setq hoge 345)

を評価する(setqは「せっとく」などと読む)。これ以降'hoge'を評価すると345が結果となる。Lispインタプリタに対して，

hoge

と入力すると，345が表示される。

(+ hoge 654)

は，もちろん999である。

変数の名前は、すでに変数として用いられていない名

前ならばなんでもよい。関数名と同じ変数名にしても構わない(が，ややこしいのでそんなことをしてはいけない)。多くの処理系ではアルファベットの大文字小文字は区別しないので'hoge' 'HOGEHOGE' 'HogeHoge'は同じ名前だとみなされるが，まれに大文字小文字を区別する処理系もある。

変数名として使えない文字も，処理系によってさまざまである。普通，括弧は使えない。C言語などと違い，'-'を含めることができる。日本産の処理系だと，マルチバイト文字(いわゆる全角文字)も使えることが多い。

では，まだsetqされていない名前を使ったらどうなるのか。'hugahuge'がまだ一度も使われていないときに，

(+ fuga 654)

とすると，まともな処理系なら「fugaには値がない」というエラーになる。

先に「関数の引数はすべて評価される」と書いた。では，初めて「(setq hoge 345)」を評価するときに'hoge'が評価されて「hogeに値がない」というエラーにはならないのか。――ならないのである。setqは1番目の引数を評価しない。setqは関数ではなく，「特殊形式(special from)」というものである。すべての引数を必ず評価するわけではない関数(のようなもの)は，関数ではなく特殊形式である。if，let，let＊，quote，setqなどが特殊形式である。

もちろん，setqの2番目の引数は評価されるので，いくらでも複雑な式を書くことができる。

(setq hoge 345)
(setq foo 654)
(setq bar (+ hoge foo))

Lispでは変数に型というものはなく，データに型がある。変数に型宣言は必要ないし，数値を代入した変数に文字列を代入したっていい。文字列は2重引用符"で囲む。

(setq hoge 345)
(setq hoge "strings")

制御と術語

まず条件分岐の雄，ifである。ifの書式は，

(if P THEN ELSE)

である。式Pの評価結果が真のときには式THENを評価し，偽のときは式ELSEを評価する。返り値は評価した式の評価結果である。

Lispにおいて，「偽」の値は'nil'で表す。nil以外の値はすべて「真」である。

(if nil 11 24)

では11は評価されず，24が評価される。

(if 3 11 24)

は11である。

「～かどうか」の判定をして「真」または「偽」を返す関数を「述語(predicate)」と呼ぶ。述語の返り値は偽'nil'もしくは真't'である。nil以外はすべて真とみなされるが，値't'を真の値の代表としている。BASIC，C言語との比較を表1に掲載する。Lispでは整数の0は真とみなされるので注意すること。

表1 真と偽の値

	偽	真	真の値の代表
BASIC	0	0以外	−1
C	0	0以外	1
lisp	nil	nil以外	t

nilやtを変数に代入することも，もちろんできる。

(setq hoge nil)
(if hoge 11 24)

述語関数はいろいろなものがある。数値どうしの関係を比較する<，<=，>，>=，あらゆる型において同一判定をするequal，型の検査をするintegerp，numberp，stringp，symbolp，nullなどなど。詳細は各処理系のマニュアルを見てほしい。述語関数の名前には最後にpが付くものが多いが，このpはpredicate (述語の意) のpである。なにかの名詞の末尾に'p'がついている場合は，引数がその前の単語であるか判定する関数だと思っておいてほしい。

真偽の値に対する演算を行うnot，and，orももちろんある。and，orはC言語と同じく，必要最小限の評価しかしない。andは引数を左から順に評価し，nilになるものがあった時点でnilを返して終了する。すべての評価結果がnil以外のときはtを返す。orは引数を左から順に評価し，nil以外のものがあった時点でtを返して終了する。すべての評価結果がnilだったらnilを返す。

・progn

(progn E0 E1 E2 ……)

prognは，ただ引数を左から順に評価する。prognの返り値は，最後に評価した式の返り値である。ifの引数のように，ひとつの式しか書けないところに複数の式を書くために用いられる。C言語の{～}，Pascalのbegin～endと同じである。

(if hoge (progn
(setq hoge 1)
(setq fuga 2))
(progn
(setq hoge 0)
(setq fuga 1)))

もしhogeがnilでないなら，hoge=1，fuga=2を代入。hogeがnilなら，hoge=0，fuga=1を代入する。

・cond

(cond (P0 E00 E01 E02 ……)

```
    (P1 E10 E11 E12 ……)
    (P2 E20 E21 E22 ……)
    ……
    (Pn En0 En1 En2 ……))
```

P0を評価し、もし真ならばE00 E01 E02 ……を評価して終了する。P0が偽ならばP1を評価し，P1が真ならE10 E11 E12 ……を評価して終了する。つまり，

1) P0～Pnは順に評価される
2) 評価結果が真になるものがあると，それに対応するEx0 Ex1 Ex2 ……を評価して終了する

ということである。condの返り値は，最後に評価した値である。P0～Pnのどれも真にならない場合は，なにも行わない。

condはC言語などのif～else if～に対応する。上記の書式例は，C言語では以下のようになる。

```
  if ( P0 ) {
     E00; E01; E02; ……;
  } else if ( P1 ) {
     E10; E11; E12; ……;
  } else if ( P2 ) {
     E20; E21; E22; ……;
 ……
  } else if ( Pn ) {
     En0; En1; En2; ……;
  }
```

前掲のprognの例をcondで書き直すと以下のようになる。ifでのELSE部は，condでは「常に真」を表す't'を条件とする。

```
  (cond (hoge (setq hoge 1)
              (setq fuga 2))
        (t (setq hoge 0)
           (setq fuga 1)))
```

・let let＊

```
  (let (V0 V1 V2 ……)
    E0 E1 ……)
```

let, let＊は局所変数を定義する。V0 V1 V2 ……に書いた変数名は，そのlet内でのみ有効な局所変数となる。E0 E1 ……はprognと同じである。

```
  (setq foo 1)  ; fooに1を代入
  foo           ; 1である
  (let (foo bar)
    (setq foo 2) ; 局所変数のfoo
    foo          ; 2である
    (setq bar 3))
  foo            ; これは1
```

変数名を書くところに'(変数名 式)'と書くと式の評価結果で初期値を与えることができる。

```
  (let ((foo 2) bar)
    foo)         ; 2である
```

この初期値の式を評価する段階では，letで生成される局所変数は使えない。

```
  (setq foo 1)
  (let ((foo 2) (bar foo))
    ……)
```

この場合のbarの値は，1である。局所変数fooではない。これが，

```
  (setq foo 1)
  (let＊ ((foo 2) (bar foo))
    ……)
```

let＊だと，barの値は2である。let＊では，初期値の式の中でも直前で定義した局所変数を使うことができる。letとlet＊の違いはこれだけである。

＊　　＊　　＊

C言語などでは文が実行されるかどうかは「通るか通らないか」という捉え方をするが，Lispでは「評価されるかどうか」と考えてほしい。

●

関数定義

●

関数を定義するのは，defunやdeなどである(処理系による)。以降，defunとして説明する。

```
  (defun FNAME (V0 V1 ……)
    E0 E1 ……)
```

関数名がFNAMEの関数を定義する。V0 V1 ……は引数を受け取る変数名である。E0 E1 ……はprognと同じで，最後に評価した結果がこの関数の返り値となる。

```
  (defun foo (a b c) ……)
```

で定義した関数fooを，

```
  (foo 3 4 89)
```

で呼び出すと，aに3，bに4，cに89が代入されてE0 E1 ……が評価されるわけである。

絶対値を返す関数absを定義してみよう。数学的な定義は，

$$|x| := \begin{cases} x & (x \geqq 0\text{のとき}) \\ -x & (\text{上記以外}) \end{cases}$$

である。これをLispで書くと以下のようになる。

```
  (defun abs (x)
    (cond ((>= x 0) x)
          (t (- 0 x))))
```

●

今回はここまで

●

次回はリストについてじっくり解説する。

今月はLispプログラムを書くための，骨格となる最低限の関数について解説した。次回はLispの基礎になるデータ構造である「リスト」についてじっくり解説する予定である。

DSP
Digital Signal Processing
(2)

320C26の基本命令とアドレッシング

Taki Yasushi 瀧　康史

DSPのプログラミングはアセンブラで行われます
構造はCPUに似ているとはいえ，DSPには特殊な概念がいくつかあります
ここでは320C26の基本命令を見ていくことにしましょう

やりたいことがすぐできない……

本編とはぜーんぜん関係ない話ですけれど。先月，ライターたるものWINDOWSもちょっとは真面目にやってみようと，手持ちのH98にDX4ODPをつけてPW805iというウィンドウアクセラレータを載っけてみました。2Mバイトのウィンドウアクセラレータなので，1152×864ドットのサイズで6万色が出ます。

それ相応には速いんですが，やっぱり，1280×960ドットサイズで16Mカラーが出ないとイマイチかなあ，PhotoShopで遊んだり印刷するには。

仕方がないので，さっさと見切りをつけて*PC-9821*Xa7を買うことにしました。いくらH98をDX4-100MHzにしたからといっても，H98のNESAバスは高速バスではなくて，総合バスなので遅いんですよね。しかも，内蔵メモリが13Mバイトで，NESAメモリで8Mバイト増設。どうにもNESAメモリはキャッシュが利いてないっぽいです。NESAメモリで動かすと，急に速度が1/10ぐらい落ちますし。

CPUが同じでも，周りが悪いとダメですね。PC-98は21シリーズになってから，バス周りがずいぶん速くなっています。Xa7はPentium75MHzですが，ちょっとイタヅラをすると，100MHzまで上げられるようですし，なにしろ値段も安いですしね。

なんでPC/AT互換機にしないんだ？ってみんなにいわれましたけど，私はときどき古いPC-98のゲームをするんです。最近，いちばんハマったゲームは，実はRevival XANADUだったりして。

メモリはとりあえず64Mバイト。これだけあれば，今年いっぱいはなんとかもつでしょうという大甘な考え。で，なにをしたいかというと，1280×1024ドットのフルカラーを使って画像処理をしたいのです。私が普段触る画像はPhotoCDの3072×2048ドットフルカラー画像です。用途が印刷ですから。このサイズになると，画像メモリはベタで18Mバイト必要です。

それでメモリが必要になりますし，WINDOWS 3.1ではシステムの都合上ダメだったりします。PC-98ではいま現在WINDOWS95が出ていないので，NTを使用しなくてはならないんだけど。NT用ディスプレイドライバがなくて困ってしまっています。マイナーではないけども，ちょっとメジャーから外れると，X68000とおんなじようなことで悩むんですね。OS/2warpでも似たようなもんみたいです。うーむ。

こんなにメモリを使うことをしている半面，2KWordで悩まなければならない世界もあります。まさにフルアセンブラなヤクザな世界。見え隠れする新Xで，そろそろアセンブラから解放されるかなとか思っていたんだけれど，やっぱり世界は甘くないようです。

最近やっとAWESOME-Xにpcmplay.xが仮対応したので，先月ローテクで紹介したDACを使用してPCMを再生して遊んでます。ただ，どうやら，DSP側のプログラムに不具合がまだあるみたいで，多少ノイズが乗るんですね。私自身，DSP側のプログラムはまだイマイチわからないので，肝心な部分は，GRAVISさんからいただいているんですけど……。残念ながら「鬼」多忙なので，あんまりよい玩具は作れそうにありません。

近況報告というと，これぐらいですね。触ってみるほどにDSPはなかなか難しいので，とりあえず今回はレジスタ周りとアドレッシングモードについて説明したいと思います。自分がやりたいことを実現するには，根性のほかないですから。

レジスタについて

レジスタを把握することは，ある程度，そのマシンのアーキテクチャを把握することにつながります。C26のレジスタは，DSPのコアに関係するレジスタと，搭載されているシリアル機能やタイマなどのレジスタとがあります。これらは68系と違い，「汎用」とはいい難いものがあります。むしろ，Z80や86系のように，1つひとつに意味があるので，その意味を把握しないとプログラムを作ることができません。詳しいことは，TMS320C25のマニュアルの3-1ページから書いてありますので，そちらを参照してください。

●アキュムレータ(ACCH，ACCL)

それぞれ，上位16ビットと，下位16ビットのアキュムレータです。2つあわせて，32ビット長になります。算術処理ユニット(ALU)の出力を格納します。

●プログラムカウンタ(PC)

プログラムメモリのアドレッシングレジスタです。プログラムメモリは64KWordなので，16ビット幅しかありません。意味は68系のそれと同等です。

●プリフェッチカウンタ(PFC)

C26がキャッシュ制御する際に使用されるものです。サイズはPCと同じ16ビットで

す。名前どおり，現在プリフェッチ中のアドレスを指し，次のプリフェッチが起こるときに内容が更新されます。ブロック転送(BLKD，BLKP)，積和演算(MAC，MACD)，テーブル読み出し/書き込み(TBLR，TBLW)命令のときデータメモリ用のアドレスにも使用されます。

●データメモリページポインタ(DP)

C26にはデータメモリとプログラムメモリがあるので，PCに対応するDPが必要になります。ただし，データメモリは64KWordありますが，16ビット幅ではなく9ビット幅です。9ビットによって，64KWord全体を128Word×512ページ(2^9)に分けるのです。そのため，ページポインタといいます。

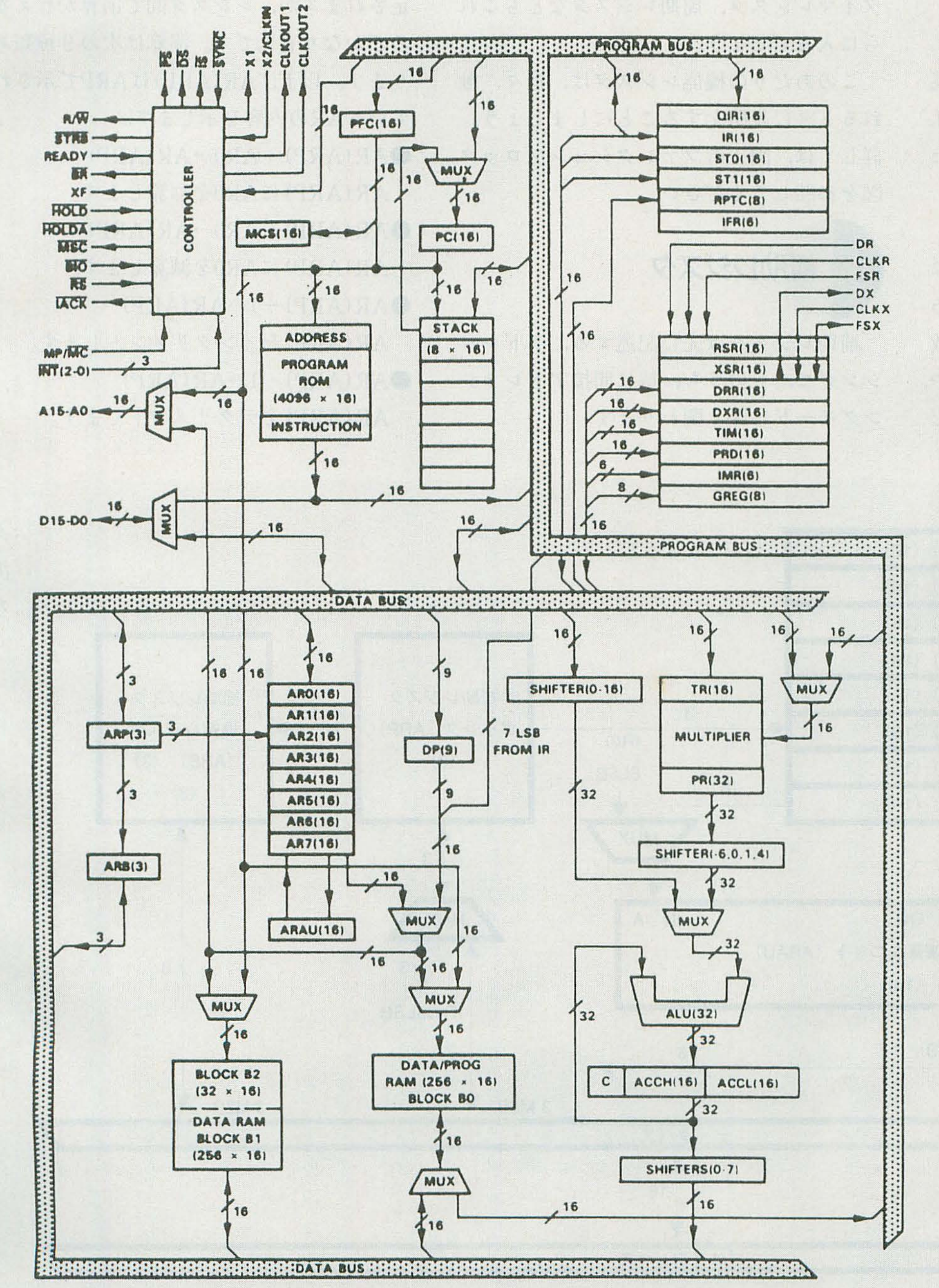
残りの128Wordをどう決定するかというと，7ビット分は命令コードの中に埋め込まれているので，それを利用してデータメモリのアドレスを最終的に決定します。つまり，128Wordの範囲でデータを収めれば，DPを更新する必要がないので，その分，高速に演算ができるということですね。86系のセグメントに近いかもしれません。

詳しくはアドレッシングモードのほうで行います。

●補助レジスタ (AR0-AR7)

強いていうならば，このAR0-AR7の8つが，汎用レジスタに近いかもしれません。ビット長は16ビットで1Word。データメモリのアドレッシングや，データの一時的格納を行います。補助レジスタ専用演算ユニット(ARAU)という，16ビット長，符

図1 TMS320C25ファンクションブロック図



注：
- ACCH ＝アキュムレータ上位
- ACCL ＝アキュムレータ下位
- ALU ＝算術論理ユニット
- AR0-AR7 ＝補助レジスタ
- ARAU ＝補助レジスタ専用演算ユニット
- ARB ＝補助レジスタポインタ退避レジスタ
- ARP ＝補助レジスタポインタ
- DP ＝データメモリページポインタ
- DRR ＝シリアルポートデータ受信レジスタ
- DXR ＝シリアルポートデータ送信レジスタ
- GREG ＝共有メモリ領域指定レジスタ
- IFR ＝割り込みフラグレジスタ
- IMR ＝割り込みマスクレジスタ
- IR ＝命令レジスタ
- MCS ＝マイクロコールスタック
- PC ＝プログラムカウンタ
- PFC ＝プリフェッチカウンタ
- PR ＝積レジスタ
- PRD ＝タイマ周期レジスタ
- QIR ＝キュー命令レジスタ
- RPTC ＝リピートカウンタ
- RSR ＝シリアルポート受信シフトレジスタ
- ST0，ST1 ＝ステータスレジスタ
- TIM ＝タイマ
- TR ＝Tレジスタ
- XSR ＝シリアルポート送信シフトレジスタ

号なしの演算装置で演算をするときにも使われます。

●補助レジスタポインタ(ARP)

補助レジスタAR0-AR7までのどれを指しているかを表すポインタです。補助レジスタが0-7までですから，ARPは当然3ビット長になります。

●補助レジスタポインタ退避レジスタ(ARB)

ARPを一時的に退避させるレジスタです。LST(ロードステータスレジスタ)命令以外，ARPがロードされるたびにARBは更新されます。LST1命令でARBがロードされると，ARPにも同じ値が書き込まれます。

●積レジスタ(PR)

乗算機からの出力を保持する32ビット長のレジスタです。上位と下位は，SPH,SPL命令を利用することにより，別々にアクセスすることができます。

●スタック (Stack：8本)

68系CPUとは違いスタックはメモリに対しては積みません。専用のレジスタがあり，それに対して積みます。そのため，数は8本と少なくなります。ACCLの内容や，データメモリの内容をスタックにPUSHしたりPOPしたりできますが，スタックポインタというものがないので，スタックのn番目という用い方はできません。

また，スタックサイズは8本を固定なので，プログラム中のサブルーチンコールは，最大でも8段までです。

●その他

プログラム中でよく利用されるのはこの程度です。あとは，機能的なレジスタであるステータスレジスタや，命令レジスタ，命令キューレジスタなどがあり，内蔵のシリアルポートを操作するためのレジスタや，共有メモリに関するレジスタがあります。タイマレジスタ，周期レジスタなどもこれらに入るでしょう。

このあたりの機能レジスタは，追々，触れるときに説明をすることにしましょう。詳しくは，図1のファンクションブロック図を参照してください。

補助レジスタ

補助レジスタは先に記述する，アドレッシングモードのうち，特に間接アドレッシングモードに深く関わります。

補助レジスタの周りは，図2のようにブロック化されています。

間接アドレッシングモードでは，図3のようにアクセスされます。図中，'>'は16進数を示します。68系では'$'を使用しますが，C26では'>'を使用すると思ってください。命令オペランドのデータメモリアドレスは，ARPで表されるARのひとつに格納され，これが示す番地のデータメモリの内容を得ることができます。わかりやすく表記するなら，AR(ARP)と書けるでしょうか。

ARAUを使用した演算は，このARPをうまく利用します。符号なしの16ビットと限定されますが，レジスタ間で演算が行えるのでかなり有効です。演算は次の9種類あります。以下，AR(ARP)はARPで示されているARの内容を示します。

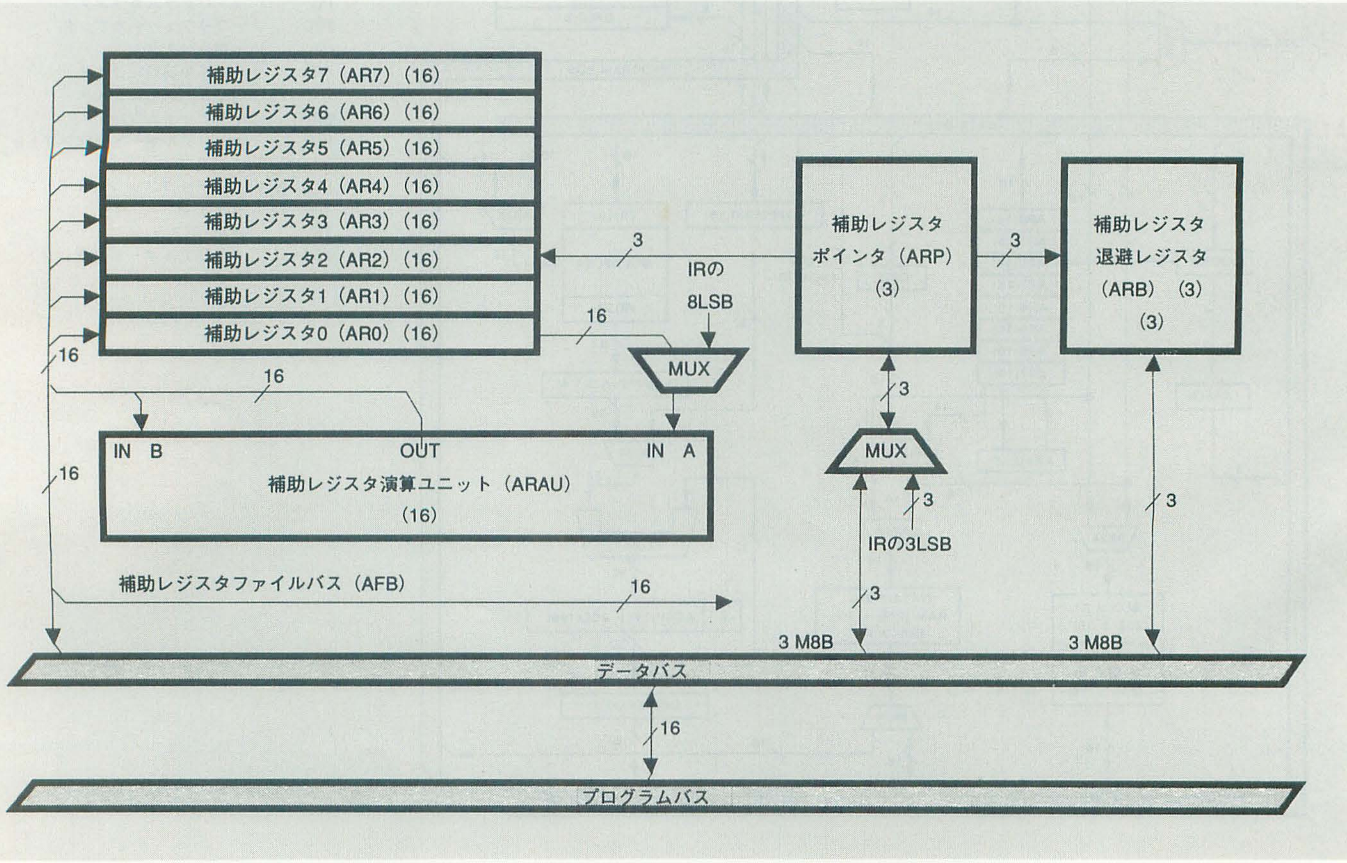
●AR(ARP)＋AR0→AR(ARP)
　AR(ARP)にAR0を加算します。

●AR(ARP)－AR0→AR(ARP)
　AR(ARP)にAR0を減算します。

●AR(ARP)＋1→AR(ARP)
　AR(ARP)をインクリメントします。

●AR(ARP)－1→AR(ARP)
　AR(ARP)をデクリメントします。

図2　補助レジスタ・ファイル



●AR(ARP)→AR(ARP)

AR(ARP)の内容を変化させません。

●AR(ARP)＋IR(7-0)→AR(ARP)

AR(ARP)に8ビットイミディエイト値（IR(7-0)）を加算します。

●AR(ARP)－IR(7-0)→AR(ARP)

AR(ARP)に8ビットイミディエイト値(IR(7-0)）を減算します。

●AR(ARP)＋rcAR0→AR(ARP)

逆方向のキャリ伝搬を行う以外AR(ARP)＋AR0と同じです。

●AR(ARP)－rcAR0→AR(ARP)

逆方向のキャリ伝搬を行う以外AR(ARP)－AR0と同じです。

図2のとおり，補助レジスタはデータメモリと直接アクセス可能なレジスタです。したがって68系のデータレジスタのような利用方法もできます。また，AR(ARP)とAR0レジスタの値を比較し，ブランチする命令も存在します。

図3　補助レジスタによる間接アドレッシングの例

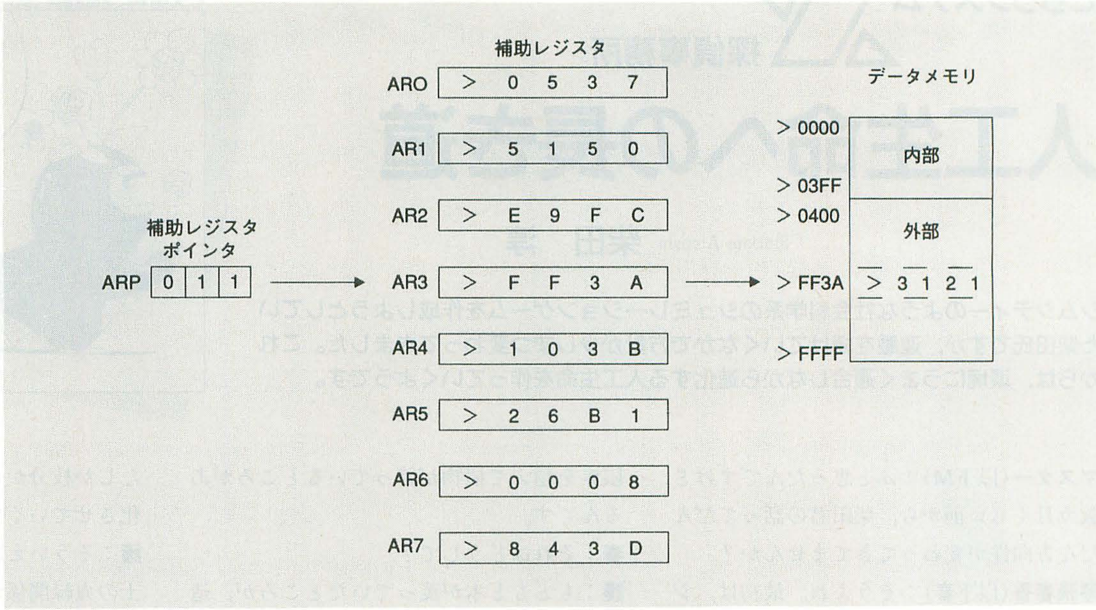
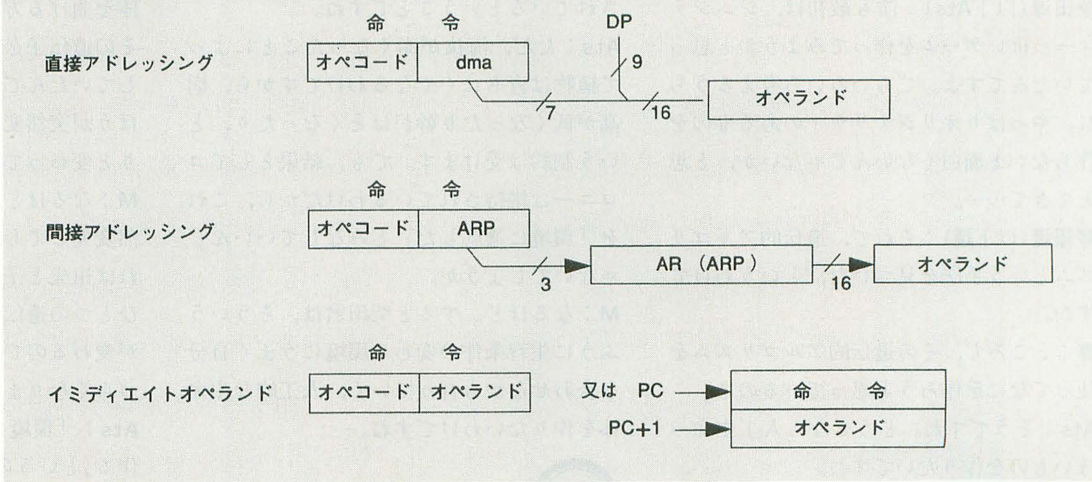


図4　命令オペランドのアドレッシング方法



アドレッシングモード

メモリアドレッシングには，直接型，間接型，イミディエイト型があります。

まずは直接アドレッシングモードの説明です。命令ワードのなかには7ビットの領域(dmaという）があり，この7ビットとデータメモリポインタ(DP)9ビット，計16ビットで64Kバイトのアドレッシング領域を決定します。直接アドレッシングは，コール命令，ブランチ命令，イミディエイトオペランド命令などに使用されます。

間接アドレッシングモードは，補助レジスタを使う方法です。68系でいう，アドレスレジスタを使う方法に似ていますが，DSPらしい面倒さで，怪しい利用方法がいくつかあります（これらは前項を参照のこと)。68系のように，アドレスはアドレスレジスタと決定されているわけではありません。命令コードの中に3ビットのARPがあり，これからAR(ARP)が指定されます。

イミディエイトアドレッシングモードはだいたい68系と同じ感じです。オペコードの中にデータを入れる方法，次のワードにデータを入れる方法があります。

詳しくは図4をごらんください。

おわりに

DSPのアドレッシングモードは，CISCの最高峰ともいえる68系CPUとは比較にならないほど難解かもしれません。正直，私も今回「これがなんの役に立つのだ？」と叫びそうになっています。実はいまだになんのためにあるんだかわからない命令もあったりして，これは当分，遊ばせてもらえそうな玩具……なような気もするんですが……。

C25のマニュアルには，FFTなどのアプリケーションも掲載されています。さすがにマニュアルだけあって難解なため，理解に苦しむことがありますが，やっぱりマニュアルがないと，どうしようもないので，ぜひ手に入れておいてください。しかし，もっともっと煮詰めねばなりませんね。

DSPへプログラムをロードするプログラムは，来月あたりに簡単なものを作成して，掲載したいなと考えておりますが，はたして……。

こちらシステムX探偵事務所

FILE-XXVII

人工生命への長き道

Shibata Atsushi 柴田 淳



illustration：T. Takahashi

シムシティーのような社会科学系のシュミレーションゲームを作成しようとしていた柴田氏ですが，連載を続けていくなかで方針が少しずつ変わってきました。これからは，環境にうまく適合しながら進化する人工生命を作っていくようです。

マスター(以下M)：ふと思ったんですけど，数カ月くらい前から，柴田君の話ってだんだん方向性が変わってきてませんか？

琴張春香(以下春)：そうよね。最初は，シムシティーみたいなゲームを作るにはどうすればいいか，という話だったはずよね。

柴田淳(以下Ats)：僕も最初は，シムシティーっぽいゲームを作ってみようかと思っていたんですよ。でもいろいろ考えるうちに，やっぱりオリジナリティのあるものを作らないと面白くないんじゃないか，と思えてきて……。

琴張護(以下護)：それで，遺伝的アルゴリズムという手法を見つけた，というわけですか。

春：ところで，その遺伝的アルゴリズムを使ってなにを作ろうと思っているの？

Ats：そうですね。どうせなら人工生命っぽいものを作りたいですね。

M：ひと言に人工生命といってもいろいろあるでしょう。

Ats：おおざっぱにいうと，ある環境があって，その環境にうまく適応するように進化していく人工生命を作ってみたいと思ってるんです。

春：それだけじゃいまいちイメージが湧かないわね。

Ats：たとえば，植物には生育できる高度限界がありますよね。

M：気温や降水量の関係で，これ以上海抜が高くなったら植物が生育できない，という限界ができるわけですね。

Ats：富士山とか日本アルプスとか，標高の高い山には基本的に木のような生命資源をたくさん必要とする植物は生えないはずなんです。ところが実際は，理論上の高度限界を超えて植物が繁っているところがあるんです。

春：それはどうして？

護：もともと木が茂っていたところが，造山活動によって海抜が高くなり，海抜が生育限界を超えても，もともとの植生が維持されているということですね。

Ats：ただ，海抜が高くなったことによって植物は育ちにくくなるわけですから，樹高が低くなったり幹がほそくなったり，という制約は受けます。でも，結果としてコロニーは維持されているわけだから，これを「環境に適応した」とみなしていいんじゃないでしょうか。

M：なるほど。すると柴田君は，そういうふうに生育条件の変わる環境にうまく自分を合わせながら育っていく，人工的な生命体を作りたいわけですね。

人工生命の三要素

Ats：と，ここまで考えついたのは，実は遺伝的アルゴリズムを試してみようと思いついたのとほぼ同時期だったんです。

春：ふーん，そうだったんだ。

護：それなら，最初に作ろうとする人工生命の概要を示しておいて，少しずつそこに向かっていく，という話の進め方のほうがスマートではなかったでしょうか。

Ats：それはそうなんですけど，その「人工生命の概要」を考えるのに結構時間がかかったんですよ。

M：どうしてですか？

Ats：3カ月前のサンプルを思い出していただけると話は早いんですが……。

春：3カ月前のサンプルプログラムって，たしか枝分かれする触手のようなものを進化させていくプログラムだったわよね。

護：そういえばあのサンプルでは，個体同士の血縁関係がどうもわかりづらかったような気がしましたが。

Ats：そうなんですよ。あのサンプルでは体を曲げる方向を遺伝子としてもっていて，その遺伝子を突然変異させて個体をコピーしていたんです。すると，遺伝子の初めのほうが突然変異した場合，個体の形ががらりと変わってしまうんです。

M：なるほど。コピーされた個体同士が突然変異しても，ある程度似た形を保っていれば祖先と子孫の関係が見えてくるけど，ひとつの遺伝子が変わっただけで大きく形が変わるのでは，確かに血縁関係はわかりづらくなりますね。

Ats：「環境に巧みに適応する人工生命を作る」というのが大前提ですけど，「見ていて楽しい人工生命」というのもそれと負けず劣らず大切だと思うんです。

春：そうじゃないと，ただの自己満足になっちゃうわよね。

護：血縁関係をわかるようにしたいならば，突然変異の際に厳しい制約をもうければすむことではないでしょうか。

Ats：突然変異の制約を厳しくすると，遺伝子に変化が現れにくくなって，今度は単調になってしまうんですよ。

M：個体が変化しすぎて血縁関係を追えないのもつまらないけど，個体に変化がなさすぎてどれも似たようなものになってもつまらない，というわけか。

Ats：いろいろ考えたんですけど，変化に富んで，しかも個体間の血縁関係を追うことのできる人工生命を作るには，次のよう

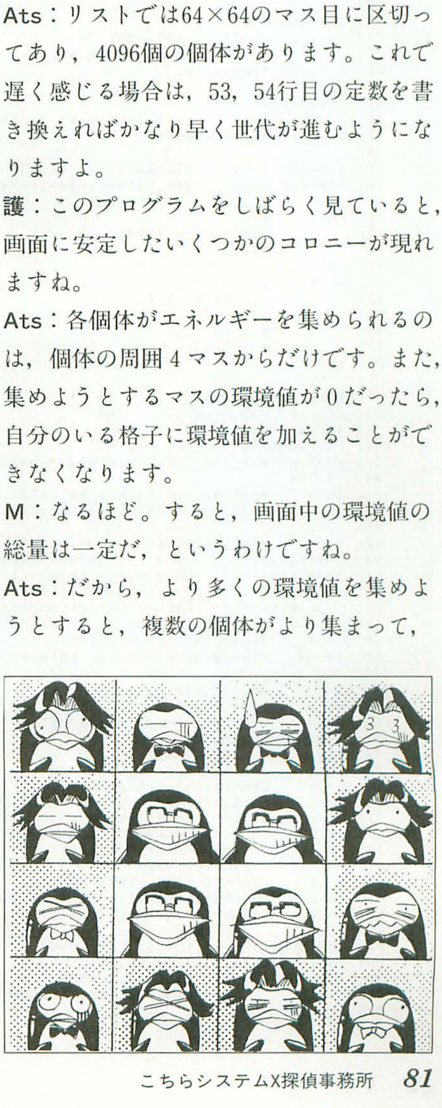
な条件を満たす必要があると思うんです。
1） 個体の行動が単純な規則に制約されている
2） 遺伝子要素の種類が少ない
3） 環境を決定する因子が少ない
護：この3つの項目を満たした人工生命というのは，とりもなおさず突然変異の制約が厳しい人工生命のことのような気がするのですが。
Ats：単純にこの項目を満たすなら，そういうことになりますね。ただ，この3つの項目を満たしてしかも見栄えをよくする方法がないわけではありません。

環境について考える

Ats：生物をミクロのレベルで見てみると，いろいろと面白いことがわかります。たとえば，動物の細胞の中にはミトコンドリアという器官があります。
護：酸素を燃焼させてエネルギーを作り出す器官ですね。細胞はミトコンドリアなしには活動することができません。
Ats：このミトコンドリアですけど，もともと細胞の中にあったわけではなくて，進化の過程で細胞内に寄生した小生物らしいということがわかっています。
春：へえ，生命活動に必要不可欠な器官がもともと寄生生物だったなんて，なんだか面白いわね。
Ats：もっと面白いのは，細胞の中でほかの器官と共生するようになった過程で，ミトコンドリアは細胞の核の奴隷になってしまったらしいんです。
M：核の奴隷になった？
Ats：遺伝子にはメジャーとマイナーがあるんですけど，細胞の構成要素の遺伝子にもメジャーとマイナーがあり，ミトコンドリアはマイナーなんです。そして，有性生殖の際には雄の遺伝子はメジャーの遺伝子しか受け継がれず，マイナーの遺伝子は雌の遺伝子から受け継がれるのです。
護：つまり，ミトコンドリアのようなマイナーの遺伝子は有性生殖には関係がなく，突然変異率が極端に低くなっているということですね。
Ats：まあ，この話は余談なのであまり深くは触れませんが，とにかく生物の細胞の中には，進化の過程で寄生したさまざまな小生物が，ときには協力しあい，ときには従属関係を結びながら目まぐるしく働いているわけです。
M：で，その話と人工生命の見栄えをよくする話のどこが関係あるんですか？
Ats：まあ，そう先を急がずに。人工生命は環境に適応するように進化するわけです。「環境」というと，森林とか海洋とか規模の大きなものを思い浮かべがちですよね。
春：そうよね。「環境」といえば，長い期間にゆっくりと変化する，いわゆる自然環境のことを考えるわよね。
Ats：でも，細胞レベルでの環境を考えてみると，雄大な自然環境とはイメージがずいぶん違ってきます。
春：どういうふうに？
Ats：たとえば，先ほどのミトコンドリアは糖を酸素で燃焼させてアデノシン三リン酸（ATP）という物質を作り出します。
護：ATPは加水分解されることでエネルギーを発生する安定な物質で，生体内の科学反応のエネルギー源として広く利用されている物質です。
Ats：で，このエネルギーを元に，DNAから合成された酵素などの代謝物質が化学反応を促進して，つまり環境を改変して，生態活動を維持していくわけです。
M：すると，ミクロのレベルで見ると，環境適応は環境改変にほかならないといいたいわけですか？
Ats：そうなんですよ。つまり，環境に適応する，という目的を受け身にとるのではなく，環境を改変しながら周囲を自分好みに作り替えてしまう行為ととらえると，人工生命の進化もかなりダイナミックになるような気がするんですが。
護：なるほど。
Ats：あと，理想としては複数の個体が，一定の環境の中で群れとして共生しながら進化していくような人工生命ができるといいかもしれません。

環境を変化させる個体

Ats：さて，今月のサンプルです。基本的に，環境値としてエネルギーのようなものをもうけて，このエネルギーを周囲からかき集めようとする個体で画面を満たします。
護：その環境値というのは，配列かなにかに保存しておくわけですね。
Ats：そうなんです。で，環境値を個体の周囲のどこからどれだけ集めるかを遺伝子によって制御します。
M：すると，個体によってはエネルギーをうまく集められないものもでてくるわけですよね。
Ats：はい。環境値がうまく集まらず，境界値以下に環境値が下がってしまった場合はその個体は死滅して，周囲の生き残っている個体がコピーされます。
春：プログラムを走らせると，画面が一面灰色になるわね。
Ats：画面では，格子ごとのエネルギー値を明るさで表現しています。各格子にはひとつずつ個体がいるので，明るい格子にいる個体は周りからエネルギーを効率的に集めていて，暗いところはあまり効率的ではない，ということになりますね。
M：画面はいくつに区切ってあるんですか？
Ats：リストでは64×64のマス目に区切ってあり，4096個の個体があります。これで遅く感じる場合は，53，54行目の定数を書き換えればかなり早く世代が進むようになりますよ。
護：このプログラムをしばらく見ていると，画面に安定したいくつかのコロニーが現れますね。
Ats：各個体がエネルギーを集められるのは，個体の周囲4マスからだけです。また，集めようとするマスの環境値が0だったら，自分のいる格子に環境値を加えることができなくなります。
M：なるほど。すると，画面中の環境値の総量は一定だ，というわけですね。
Ats：だから，より多くの環境値を集めようとすると，複数の個体がより集まって，



連携して働かなければならないんです。

春：ふーん。それで明るい部分がかたまっている，というわけね。

Ats：あと欲をいえば，個体間でフィードバックの仕組みを取り入れる工夫ができるといいんですけどね。

M：フィードバックというと？

Ats：つまり，ある個体の行動の結果が，ほかの個体の行動に影響を与えるような仕組み，といったらわかるかな。

護：実際の生命活動でも，ある代謝物質を作りすぎると，酵素のもとになるRNAの生産が抑制されたりといった仕組みがあるようですね。

Ats：そうなんですよ。このフィードバックが生命活動を維持するうえでかなり大きな役割を果たしているらしいんです。で，人工生命にもこのフィードバックを取り入れられれば，もう少し見栄えがよくなるはずなんです。このあたりが今後の課題といったところかな。　　　　(つづく)

リスト

```
/*
		環境を改変する個体 任意のキーで終了
		(注) 8月号掲載のリストUtilities.cが必要
*/

#include <stdio.h>
#include <stdlib.h>
#include <stddef.h>
#include <string.h>

#define _GetParam(ptr)	((*(ObjectListPtr)ptr).params)
#define _GetSig(ptr)	((*(ObjectListPtr)ptr).object_sig)

#include "Utilities.c"
/*8月号掲載のリストをインクルード*/

typedef struct	{
		void	*blockFunc,*get_paramFunc,
				/*実行関数，パラメータ取得関数のポインタ*/
				*next_block;
				/*次の遺伝子要素のポインタ*/
		long	x,y,condition;
} Blocks,*BlocksPtr;
/* 遺伝子要素の構造体 */

typedef struct	{
		long	x,y;
} blockPar,*blockParPtr;

typedef struct	{
		void*	next_act;
		long	cond,no_con,x,y;
} Param,*ParamPtr;

typedef struct	{
		long	par1,par2,par3,depth;
} GPParam,*GPParamPtr;

typedef long	blocks(void*,blockParPtr);
typedef long	get_param(void*,long depth,GPParamPtr);

enum	{
		junction = 1000,
		block
};
enum	{
		add_right = 0,
		add_left,
		add_up,
		add_low
};

#define map_width		64
#define map_height		64
#define center_val		64
#define minimam_val		1
#define max_ind	map_width*map_height

long	map[map_width][map_height],
		map2[map_width][map_height];

long	RegisterObject(void*,void*,void*,long,long);
ObjectListPtr	ConstructObject(long,void*);
ObjectListPtr	CopyObject(ObjectListPtr);
void	DeleteObject(ObjectListPtr);
ObjectListPtr	GetLastObject(ObjectListPtr);
void	StandardCreator(ObjectListPtr,void*);
void	StandardDistructor(ObjectListPtr);
void*	StandardCopier(ObjectListPtr);

void	Initialize();
void	DoBlocks();
void	DrawRect(long,long,long);
void	Mutate();
void	InitMap(void);
long	AddInd(ObjectListPtr ob);
void	Change(ObjectListPtr ob);
void	Exchange(ObjectListPtr ob1,ObjectListPtr ob2);
ObjectListPtr	Getblock(ObjectListPtr,long);
void	Createblock(void* this,void* param);
void	Distructblock(void* this);
void*	Copyblock(void* this);
long	ablock(void*,blockParPtr);
long	aGetParam(void*,long,GPParamPtr);
long	AddMap(long,long,long);

ObjectListPtr	IndList[max_ind];
long	total = 1,wx,wy,ox,oy,wpos,
			foots[max_ind],
			dep[max_ind],IndCount;

void Initialize(void)
{
long	i;
		allmem();
		screen(1,3,1,1);
		for( i = 0; i != max_ind; i++ ) {
				IndList[i] = NULL;
		}
		InitMap();
}

void	main()
{
long	i,x,y,siz = sizeof(Blocks);
Param	tmpP;

		Initialize();
/* Registering objects */
		RegisterObject(Createblock,Distructblock,
						Copyblock,siz,block);
		tmpP.next_act = NULL;
		tmpP.cond = add_right;
		i = 0;
		for( y = 0; y != map_height; y++ ) {
				for( x = 0; x != map_width; x++ ) {
						tmpP.cond = rand()*4/RAND_MAX;
						tmpP.x = x;
						tmpP.y = y;
						IndList[i] = ConstructObject(block
,&tmpP);
						i++;
				}
		}
		do {
				DoBlocks();
				Mutate();
		}while( !kbhit() );
		return;
}

void	DoBlocks()
{
long	i,x,y;
blockPar		ap;
BlocksPtr		tmp;
GPParam gp;
		IndCount = 0;
		i = 0;
		for( y = 0; y != map_height; y++ ) {
				for( x = 0; x != map_width; x++ ) {
						ap.x = x;
						ap.y = y;
						if( IndList[i] != NULL ) {
								tmp = (BlocksPtr)((*IndLis
t[i]).params);
								((blocks*)(*tmp).blockFunc
)(IndList[i],&ap);
						}
						i++;
				}
		}
		for( x = 0; x != map_width; x++ ) {
				for( y = 0; y != map_height; y++ ) {
						DrawRect(x,y,map[x][y]);
				}
		}
}

void	DrawRect(long x,long y,long val)
{
long	col;
		if( val > 127 ) {
				val = 127;
		}
		col = rgb(val/4,val/4,val/4);
		fill( x*8,y*8,x*8+7,y*8+7,col );
}

void	Mutate()
{
long	i = 0,x,y;
		for( y = 0; y != map_height; y++ ) {
				for( x = 0; x != map_width; x++ ) {
						if( map[x][y] < minimam_val &&
								i != 0 ) {
								DeleteObject(IndList[i]);
								IndList[i] = CopyObject(In
dList[i-1]);
						}
						if( rand() < RAND_MAX/60 )
								Change(IndList[i]);
```

```
			if( rand() < RAND_MAX/30 && i != 0
)
				Exchange(IndList[i],IndLis
t[i-1]);
			i++;
		}
	}
}

void	Change(ObjectListPtr ob)
{
long	i;
GPParam gp;
ObjectListPtr	op1,op2,op3;
BlocksPtr	tmpA;
Param	tmpP;
	tmpA = (BlocksPtr)((*ob).params);
	gp.depth = 0;
	((get_param*)(*tmpA).get_paramFunc)(ob,9999,&gp);
	i = gp.depth;
	if( i < 2 )  return;
	i = (double)rand()/RAND_MAX*(i-2)+1;
	op1 = Getblock(ob,i-1);
	op2 = Getblock(ob,i);
	op3 = Getblock(ob,i+1);
	(*(BlocksPtr)((*op2).params)).next_block = NULL;
	DeleteObject(op2);
	tmpP.next_act = op3;
	tmpP.cond = 0;
	tmpP.no_con = -1;
	if( rand() > RAND_MAX/2 ) {
		op2 = ConstructObject(block,&tmpP);
	} else {
		op2 = ConstructObject(junction,&tmpP);
	}
	(*(BlocksPtr)((*op1).params)).next_block = op2;

}

void	Exchange(ObjectListPtr ob1,ObjectListPtr ob2)
{
long	i;
GPParam gp;
ObjectListPtr	opA1,opA2,opA3,opB1,opB2,opB3;
BlocksPtr	tmpA,tmpB;
	tmpA = (BlocksPtr)((*ob1).params);
	gp.depth = 0;
	((get_param*)(*tmpA).get_paramFunc)(ob1,9999,&gp);
	i = gp.depth;
	tmpB = (BlocksPtr)((*ob2).params);
	((get_param*)(*tmpB).get_paramFunc)(ob2,9999,&gp);
	if( i > gp.depth ) {
		i = gp.depth;
	}
	i = (double)rand()/RAND_MAX*(i-1)+1;
	opA1 = Getblock(ob1,i-1);
	opA2 = Getblock(ob1,i);
	opA3 = Getblock(ob1,i+1);
	opB1 = Getblock(ob2,i-1);
	opB2 = Getblock(ob2,i);
	opB3 = Getblock(ob2,i+1);
	(*(BlocksPtr)((*opA1).params)).next_block =
						opB2;
	if( opA2 != NULL ) {
	(*(BlocksPtr)((*opA2).params)).next_block =
						opA3;
	}
	(*(BlocksPtr)((*opB1).params)).next_block =
						opA2;
	if( opB2 != NULL ) {
	(*(BlocksPtr)((*opB2).params)).next_block =
						opB3;
	}
}

ObjectListPtr	Getblock(ObjectListPtr op,long depth)
{
ObjectListPtr	tmpA;
	tmpA = op;
	while( depth > 0 ) {
		tmpA = (*(BlocksPtr)((*tmpA).params)).next
_block;
		if( tmpA == NULL ) {
			break;
		}
		depth--;
	}
	return( tmpA );
}

void	InitMap()
{
long i,j;
	for( i = 0; i != map_width; i++ ) {
		for( j = 0; j != map_height; j++ ) {
			map[i][j] = center_val;
			map2[i][j] = 0;
		}
	}
}

void	Createblock(void* this,void* param)
{
	(*(BlocksPtr)_GetParam(this)).blockFunc =
		ablock;
	(*(BlocksPtr)_GetParam(this)).get_paramFunc =
		aGetParam;
	(*(BlocksPtr)_GetParam(this)).x =
		(*(ParamPtr)param).x;
	(*(BlocksPtr)_GetParam(this)).y =
		(*(ParamPtr)param).y;
	(*(BlocksPtr)_GetParam(this)).condition =
		((double)rand()/(RAND_MAX+1))*4;
	if( (*(ParamPtr)param).next_act != NULL ) {
		(*(BlocksPtr)_GetParam(this)).next_block =
			(*(ParamPtr)param).next_act;
	}
}

void	Distructblock(void* this)
{
	if( (*(BlocksPtr)_GetParam(this)).next_block != NU
LL ) {
		DeleteObject((*(BlocksPtr)_GetParam(this))
.next_block);
	}
	StandardDistructor(this);
}

void*	Copyblock(void* this)
{
ObjectListPtr	newObject;
	newObject = StandardCopier(this);
	if( (*(BlocksPtr)_GetParam(this)).next_block != NU
LL ) {
		(*(BlocksPtr)(*newObject).params).next_blo
ck =
			CopyObject((*(BlocksPtr)_GetParam(
this)).next_block);
	}
	return( newObject );
}

long	ablock(void* this,blockParPtr	parm)
{
BlocksPtr	tmp;
long	tx,ty,ret = 0;

	tmp = (*(BlocksPtr)_GetParam(this)).next_block;
	tx = (*parm).x;
	ty = (*parm).y;
	switch( (*(BlocksPtr)_GetParam(this)).condition )
{
		case add_right :
			AddMap(tx,ty,1);
			AddMap(tx+1,ty,-1);
			break;
		case add_left :
			AddMap(tx,ty,1);
			AddMap(tx-1,ty,-1);
			break;
		case add_up :
			AddMap(tx,ty,1);
			AddMap(tx,ty-1,-1);
			break;
		case add_low :
			AddMap(tx,ty,1);
			AddMap(tx,ty+1,-1);
			break;
	}
	if( tmp ) {
		ret = ((blocks*)
			(*(BlocksPtr)_GetParam(tmp)).block
Func)(tmp,parm);
	} else {
		return(0);
	}
	return( ret );
}

long	aGetParam(void* this,long depth,GPParamPtr gp)
{
BlocksPtr tmp;
	if( depth == 0 ) {
		(*gp).par1 = (*(BlocksPtr)_GetParam(this))
.condition;
		(*gp).par2 = block;
		return( 0 );
	}
	depth--;
	(*gp).depth++;
	tmp = (*(BlocksPtr)_GetParam(this)).next_block;
	if( tmp ) {
		((get_param*)
		(*(BlocksPtr)_GetParam(tmp)).get_paramFunc
)(tmp,depth,gp);
	}
	return( 0 );
}

long	AddMap(long x,long y,long val)
{
	if( x > map_width ) {
		x -= map_width+1;
	}
	if( x < 0 ) {
		x += map_width;
	}
	if( y > map_width ) {
		y -= map_height+1;
	}
	if( y < 0 ) {
		y += map_height;
	}
	if( val < 0 && map[x][y] == 0 ) {
		return;
	}
	map[x][y] += val;
}
```

SX-WINDOW ver.3.1 開発キット

Tamura Kento 田村 健人

8月下旬に「SX-WINDOW ver.3.1 開発キット」が発売された。5インチフロッピーディスク1枚とCD-ROM1枚付きの書籍である。この書籍は以下の2つの役割を持っている。

・SX-WINDOWのソフトを開発するための環境を補強する

・SX-WINDOWのフリーソフトウェア/シェアウェアを配布する

開発環境の補強

開発キットという名称はついているが，この書籍だけでSX-WINDOWの開発環境がすべて揃うわけではない。すでに開発環境を持っている人が，それを補強するためのソフトウェアが収録されている。SX-WINDOW ver.3.1の開発を行うためにはこのキットとWorkroomが必要である。

以下の3つが付属フロッピーディスクに入っている(CD-ROMにも入っている)。

●SX31KIT

SX-WINDOW ver.2.0の開発環境しか提供しなかったWorkroom SX-68Kの開発環境を，SX-WINDOW ver.3.1に対応させるためのヘッダとライブラリである。

●libsxc

libcを，SX-WINDOWの開発にも使用できるようにしたものである。生のXCライブラリやlibではSX-WINDOW開発に使える関数はごく一部に限られていたが，libsxcではほぼすべての関数を利用できる。ソースを書くときに特に意識しなくても，リエントラントな実行ファイルを作成できる。libsxcはlibcを含んでいるので，そのまま最新のlibcとしても利用できる。



●callno header

C言語でのSXコール呼び出しを，ライブラリを通さずに行うためのヘッダファイルである。これを用いるだけで，ソースの変更はほとんどなしにプログラムをコンパクトで・高速にすることができる。Workroomのヘッダでも，「追補版SX-WINDOWプログラミング」のヘッダでも利用できる。libcの__IOCS_INLINE_，__DOS_INLINE_と同じようなものである。

開発関連で上記以外に本に印刷されているのは，SX-WINDOWプログラミングの基礎的な解説，ver.3.1でのプログラミング例，SX-WINDOW ver.3.1までの全SXコールのリファレンスマニュアルである。

プログラミング例では，階層ウィンドウ，静止画の伸長と圧縮，動画(CGAファイル)の再生と作成の解説がされている。IVM用の画像モジュールの作成方法には触れられていないが，CD-ROMのほうにフリーソフトのモジュールのソースがあるのでそれを参考にすればなんとかなるだろう。

SXコールリファレンスは残念ながらFSX.X，IFM.X，IVM.Xのコールだけが対象で，SXCON.Xに関しては載っていない。インラインのかな漢字変換やコマンドシェル上のプログラムの実行に関しては謎のままである。これらの機能は「シャーペン」という一アプリケーションの機能である，ということなのだろう。また，Workroomと同じく未公開コールについては一切載っていない。

ソフトの配布

付属のCD-ROMにはフリーソフトウェアによるさまざまなSXアプリケーションが収録されている。これには「集大成」という言葉が実によくにあう。QuTERM，ViSON，grrootのような超メジャーソフトから，mvsi.xのようななんの役にも立たないものまで200本弱のソフトがCD-ROMに収録されている。添付フロッピーディスクには計測技研のCD-ROMドライバの機能限定版とフリーソフトのSCSIドライバSUSIEが入っているので，CD-ROMドライバを持っていなくても大丈夫である。これらのドライバでも認識できないようなひねくれたCD-ROMドライブを持っていた場合は，自分の不幸を呪うしかない。

昨年秋発売の「Free Software Selection Vol.2」は収録作品を募集する形態だったが，今回はNIFTY-ServeとNetwork SXNG*1にあるソフトの作者に収録許可を得るかたちで収集されており，収録内容は充

SX-WINDOW ver.3.1開発キットの不具合

8月末に弊社から発売されました書籍「SX-WINDOW ver.3.1開発キット」添付のフロッピーディスクのフォーマットが作業上の手違いから1.2MバイトのIBMフォーマットになっていました。

Human68k ver.3.0以降をお持ちの方の場合，FDDEVICE.Xを組み込むことで問題なく添付FDをご利用いただけます。

手順はCONFIG.SYSに，

DEVICE=¥SYS¥FDDEVICE.X

のような1行を加えるか，起動時のコマンドラインから，

A>FDDEVICE

と打ち込んで起動させます。これで添付FDをアクセスすることができるようになります。FDDEVIE.Xに関して，詳しくはHuman68kのマニュアルをお読みください。

Human68k ver.2.xxしかお持ちでない場合，フリーソフトウェアの9scdrv.xを使用すると，同様にご利用いただけるようになります(9scdrv.xはNIFTY-Serveなどパソコン通信で入手することが可能です)。

また，Human68k ver.3.0以降をお持ちでなく，かつ，9scdrv.xを入手できない場合，以下の宛先まで購入されたFDをお送りいただければHuman68kフォーマットのFDをお送りいたします。

〒103 東京都中央区日本橋浜町3-42-3
ソフトバンク株式会社出版事業部
ハードウェア活用書籍編集部
「SX3.1開発キットFD係」

実している。多くのソフトがソースと共に収録されているので，プログラミングの参考になると思う。

CD-ROMには基本的にアーカイブが解かれた状態で置いてあるが，ビル・ゲイツとISOの呪いによりファイル名は8.3形式である。よって，CD-ROM上では実行できないソフトが多い。付属のインストーラがハードディスクにコピーするときに本来の名前に戻してくれる。

たとえば，IVM用のモジュールだけでもMAG/MAKI/Pi/CUT/GIF/PIC2/SC8/ZAUと，とても充実している。

手前味噌であるが，本誌付録ディスク「もみじ狩りPRO-68K」に収録されたシャーペン外部コマンドとarlkのバグフィクス版もある。

gcc(based on 1.42)/HAS/hlkの最新版，gcc-2.6.3ベースのC/C++/Objective-Cコンパイラやlibg++-2.6.2の移植版も入っているので，SX-WINDOWとは関係なくHuman68kの開発環境を充実させることもできる。NEmacsとMuleはSX-WINDOWでもコマンドシェルでも実行できる。

*1 「たった2カ所？」と思われるかもしれないが，SX-WINDOWのソフトはこの2カ所でほとんど揃ってしまうらしい。

まとめ

SX-WINDOWを使っている人にとっては極上の一品である。SX-WINDOWを使っていなくても，Human68kの開発環境とMuleのために買ってしまってもいいかもしれない。

ぜひCD-ROMから滲み出るSXerの情熱と怨念を感じてほしい。

SX-WINDOW ver.3.1 開発キット

吉沢正敏/牛島健雄/西田文彦/小浜純著

5,800円（税込み）　ソフトバンク刊

CD-ROMに収録されているソフト一覧

APL/3DGRAPH	3D Graph
APL/CALEN	カレンダー
APL/CDDATA	ＣＤの管理とラベル作成支援
APL/CGRAPH	グラフ作成
APL/DATAUTY1	C-Graph SXCalc用 データ間引き
APL/DATAUTY2	C-Graph SXCalc用 非線形近似
APL/DENTAKU	電卓
APL/GMENU	クリップボードのデータをグラフ化
APL/GNUPLOT	グラフ作成ソフト gnuplot
APL/GRAPH	C-Graph SXCalc用 グラフモジュール
APL/GRAPH2	C-Graph SXCalc用 グラフ描画TOOLS(2)
APL/MCZ	Zeitフォントマージ
APL/MEMG	メモリメーター
APL/MVSI	SX-WINDOWを一部アニメ化する
APL/POSTSX	郵便番号検索
APL/SC55ED	SC-55mkIIエディタ
APL/SXCALC	表計算(シェアウェア)
APL/SXDB	データベース
APL/SXMEMO	メモアクセサリ
APL/SXPM	負荷計測
APL/SXTOKEI	メモリ消費の少ない時計
APL/TOKEI	アラーム機能付鳩時計
APL/XTRANTE	表計算
APL/ZEITOR	フォントエディタ
ARC/SXGZIP	GNU gzip-1.2.4 (ViSON対応)
ARC/SXTAR	tar展開 (ViSON対応)
ARC/SXXX	lzh展開 (ViSON対応)
ARC/VISON	アーカイブビュア
DATA/CARD	フリーリソースデータ(トランプ)
DATA/PAIMIN	麻雀牌リソース (縮小)
DATA/PAISTD	麻雀牌リソース (標準)
DATA/SXLOGO	SXPanic用サンプルデータ
DATA/UNO	フリーカードリソース
DESKTOP/EYE	目玉
DESKTOP/MISA	みさちゃん
DESKTOP/SESS	超簡易スクリーンセーバー
DESKTOP/SXMODULE	DA集スクリーンセーバ用モジュール3つ
DEVELOP/ARLK	resource linker
DEVELOP/CALLNO	SXCALLのインラインヘッダ(添付FDに収録)
DEVELOP/CCCV	逆ＣＣＶコンバータ
DEVELOP/CPLK	CGAファイルのPCMを加工
DEVELOP/GCC	GNU C Compiler 1.28 (based on 1.42)
DEVELOP/GCC2	gcc-2.6.3 基本キットとCコンパイラ
DEVELOP/GCC2GPP	gcc-2.6.3 C++コンパイラ
DEVELOP/GCC2OBJC	gcc-2.6.3 Objective-Cコンパイラ
DEVELOP/HAS	ハイスピードアセンブラ
DEVELOP/HLK	ハイスピードリンカ
DEVELOP/HLKP2	hlk-3.01 .quad対応 *非公式* 差分
DEVELOP/LIBGPP	libg++-2.6.2 C++用クラスライブラリ
DEVELOP/LIBSXC	libsxc 1.1.32 (添付FDに収録)
DEVELOP/LVCV	LVファイルをCGAにする
DEVELOP/MENUDESI	メニューリソース作成
DEVELOP/OAR	オブジェクトアーカイバ
DEVELOP/RCONV	*.xを*.rに変換
DEVELOP/RSCV	Resource Viewer
DEVELOP/SX31KIT	SX v 3.1 開発キット(添付FDに収録)
DEVELOP/XDB	db/gdbでのデバッグを可能にする
DOC/INSIDE	開発の際に便利なオンラインドキュメント
DOC/VMINST	IVMモジュールインストールスクリプト
EDITOR/MAPEDIT	BGマップエディタ
EDITOR/MULE	Human/SX 両用Mule
EDITOR/NEMACS	Human/SX 両用NEmacs
EDITOR/SXNG	Nemacs ライクなエディタ
EXT/2LINE	長いアイコン名を折り返す
EXT/ACTJP	アクティブウィンドウを表示画面に収める
EXT/ADI	ディレクトリ表示窓拡張
EXT/BEEPCHG	ビープ音変更
EXT/CGBFD	キャンバス/CGビジョンをClickMenu風に
EXT/CLICK	メニューをクリックで選択
EXT/DIAGE	複数デスクトップ
EXT/EXC	Oh!X外部コマンド bug fix
EXT/EXTDRAG	ViSONのextdragの障害回避
EXT/FCMPL	ファイル名補完機能を付加
EXT/FIXMV	マウント時の不都合を回避
EXT/FIXSEMB	SX3.01排他制御不具合修正
EXT/HCDAD	Human>CDオーディオを高速化する
EXT/HENWIN2	漢字変換window
EXT/IWIDTH	アイコンの間隔を変更
EXT/KAISO	ポップアップメニューをハデに・階層化
EXT/MPERI	FSX.X マルチピリオド対応化
EXT/SADJUST	画面モード設定
EXT/SETDTOP	DIRDTOP.SXのパスを設定する
EXT/SILENT	ウインドウを並べ替えずにアクティベート
EXT/SNAP	マウスポインタをボタンに移動
EXT/SXERROR	エラーリカバリ
EXT/SXWS	もうひとつのページアイコン
EXT/WINLOC	ウィンドウ出現位置を指定
EXT/WINSEL	ウィンドウをキーコントロール
EXT/WORKSPC	複数デスクトップ
FILE/DINFO	ドライブ情報表示
FILE/DRVINFO	ドライブ情報表示
FILE/FILEINFO	ファイル情報表示
FILE/FINFO	ファイル情報表示
FILE/FMEMO	ファイルメモ
FILE/SXINFO	ＳＸ情報表示
GAME/GNUCHESS	Gnu Chess
GAME/GOLFEDIT	コースエディタ for SxGOLF
GAME/GOLFPCM	SxGOLF 効果音キット
GAME/PITMAN	SX-PITMAN
GAME/SIRTET	SX-Sirtet
GAME/SXCHN	麻雀牌パズル『SX青海』
GAME/SXGOLF	SX GOLF
GAME/SXHONG	SX香港
GAME/SXJUNKS	SX用ゲーム & アクセサリ
GAME/SXKRON	クロンダイク
GAME/SXMIN	マインスイーパ
GAME/TATRIS	SX-Tatris
GRAPH/GATELIER	6万色グラフィックエディタ
GRAPH/GHOSTSCR	Human>Ghostscript PostScript表示
GRAPH/GHOSTSSX	Ghostscript PostScript表示
GRAPH/GLMPAT	GLM for IVM
GRAPH/GRROOT	6万色背景
GRAPH/GRSEL	grroot 用の画像選択
GRAPH/HEKIDO	壁紙動画
GRAPH/HEKITM	壁動玉々
GRAPH/IVMAG	MAG for IVM
GRAPH/IVMAKI	MAKI for IVM
GRAPH/PIVM	Pi for IVM
GRAPH/SXPANIC	SX で PANIC を再生
GRAPH/SXPICONV	PIX<->Pi コンバータ
GRAPH/SXPIXPI	PIX, Piローダ
GRAPH/VMCUT	CUT for IVM
GRAPH/VMGIF	GIF(sm) for IVM
GRAPH/VMP2	PIC2 for IVM
GRAPH/VMSC8	VMSC8 for IVM
GRAPH/VMZAU	ZAU for IVM
GRAPH/XBMP	Human>BMPローダ
LANG/SXLISP	xlisp 2.1 の SX 版
LIB/COMHEAD	WILL氏の共通ヘッダ
LIB/GUGESLIB	Guges氏の開発環境
LIB/ICONWDEF	アイコンウィンドウ定義関数
LIB/IOMAINP	libg++-2.6 X6_02 の iomanip.o
LIB/LIBCOC	SX/Human 共用バイナリ用ライブラリ
LIB/LIBSEL	セレクショライブラリ
LIB/MOCDEF	立体ボタンリソース
LIB/MSX	METHOD SX library
LIB/PICBTTN	ピクチャボタンリソース
LIB/SXHEL	Human->SX 移植ライブラリ
LIB/SXTOOL	SXアプリ開発用ツールキット
LIB/TSSLIB	普通のサブルーチンを疑似マルチタスク
LIB/UPDOWN	数値調整ボタン用ライブラリ
LIB/WORKGCC	Workroom+gccで、OBJRを作製する開発環境
MUSIC/ADPPLAY	ただPCMを鳴らすだけ
MUSIC/SXCDP	CD-PLAYER
MUSIC/SXWAV	WAVファイルプレイヤー
MUSIC/SXZC	Z-MUSIC/RC player
MUSIC/SXZCSUP	sxzc support softwares
MUSIC/WORDMK	CD-PLAYER 用歌詞表示プログラム
SX110/SHUTTER	画面退避/表示
SX110/SXPT	SX 1.10 の不具合?修正
TERM/BPLUS	B-Plusプロトコル
TERM/CZCOMP	コンピュータ画面にする
TERM/GNUS	GNUS
TERM/MLINK	mlinkプロトコル
TERM/NR	News Reader
TERM/QINT	マウスの割り込み負荷を軽減する
TERM/QUTERM	LISP搭載ターミナルソフト
TERM/QUTERMS	2Mメモリでも起動する QuTERM mini
TERM/SXQV	Quick-VANプロトコル
TERM/SXRSX	通信中の文字落ちを防ぐように努力する
TOOL/AUTOCASE	ファイル名自動大文字化
TOOL/CALCMAN	OPT1+OPT2で電卓などを起動
TOOL/CALCSX	計算機
TOOL/CANVAS	適切な viewer を起動
TOOL/CHDIR	DOSのカレントディレクトリをSX上で変更
TOOL/CLNOTE	便利なラウンチャ
TOOL/CLSEWITH	ほとんど全クローズ
TOOL/COPYBACK	ファイルバックアップ
TOOL/CRTCTRL	CRTCのレジスタを変更して画面モードを作成
TOOL/CUPMEN	カップめんタイマー
TOOL/DOC	Human>ドキュメント閲覧
TOOL/DV	ディレクトリビューア
TOOL/EMAGEN	メニュー以外でSXシェルから脱出
TOOL/ENZAN	グラフ描画機能付き関数電卓
TOOL/F2SC	ファイルからスクラップへ
TOOL/GARBAGE	未開放メモリを表示・開放
TOOL/GCOPY	画面イメージをクリップボードに転送
TOOL/HANTRAY	汎用ファイルトレイ
TOOL/HISCLIP	ヒストリ付クリップボード
TOOL/HZAN	メモリの残量を表示
TOOL/IVMINFO	IVMリソース情報を表示
TOOL/KONPOI	ファイルをクリーナへ移動
TOOL/MFOCK	複数の実行ファイルを同時に起動
TOOL/MINI	縮小デスクトップ
TOOL/MOJI	文字列検索
TOOL/NDS	新規ファイルを作る
TOOL/NEWPATH	パス名取得
TOOL/NSMAN	Human>マニュアル閲覧 SXManで利用
TOOL/NURU	さまざまなフィルタリング
TOOL/OPENDIR	ディレクトリオープナ
TOOL/PATH	パス名取り出し
TOOL/PICCON	画像ファイルをアイコン化
TOOL/PICTDRAW	PICT → DRAW 変換
TOOL/PSKILL	シャーペンのコンソール上でps&kill
TOOL/PT4GET	シャーペンファイルからアイコンを取得
TOOL/REGSEA	シャーペン正規表現検索
TOOL/REN	ファイル名を変更
TOOL/RMIVM	IVMモジュール削除
TOOL/ROOTCOPY	grroot画像をスクラップにコピー
TOOL/SCOPY	COPYキーで画面をスクラップにコピー
TOOL/SEND	コマンドラインからSXBasic形式通信
TOOL/SMAUTO	ストゥリーム対応タスクラウンチャ
TOOL/SMBACK	ストゥリームの画像で壁紙を変更
TOOL/SMCUT	ストゥリームの画像から一部を切り出す
TOOL/SMGAL	ストゥリーム対応グラフィックローダ
TOOL/SMGF	ストゥリーム対応の65536色画像ビューア
TOOL/SMSCST	ストゥリーム⇔スクラップコンバータ
TOOL/SMSTREAM	ストゥリーム機能を提供するマネージャ
TOOL/SMSXG2T	グラフィック→テキスト変換
TOOL/SXAD	ディスク挿入でディレクトリオープン
TOOL/SXBDIF	Bdif.x のSX版
TOOL/SXBJ	ESC/P, CZ 用プリントアウトツール
TOOL/SXBUP	Bup.xのSX版
TOOL/SXCRLF	改行コード変換
TOOL/SXGREP	文字列検索
TOOL/SXJIS	漢字コード変換
TOOL/SXMAN	NSman 形式ドキュメントを見る
TOOL/SXMIC	ish decoder
TOOL/SXMODE	16色 or 65536色モードを表示
TOOL/SXMOMO	1ドライヴでMOをコピー
TOOL/SXMP	マウス座標を表示
TOOL/SXPFM	performance測定&cache/wait設定
TOOL/SXSED	sed
TOOL/SXSH	コマンドラインの簡易シェル
TOOL/WINMAN	窓を操る for CLICKNOTE
TOOL/WINMOV	キーボードで窓を動かす
TOOL/WINTHIEF	ウィンドウをクリップボードへ

マルチシンクモニタ

PC-TM151

Taki Yasushi 瀧　康史

高解像度のSX-WINDOWも楽しんでみたい。だけども，15kHzが映るモニタは残しておきたい。すでにゲーム機だって15kHzだし，ゲーム基板だってある。X680x0のゲームだってときどき15kHzでやりたくなるし，テレビやビデオやLDだってある。「ハイスキャンのモニタとテレビを別々に買いなさい」といいたいところですが，住宅事情もあるでしょう。そこまでモニタにこだわってないという人もいるかもしれません。そんな人にぴったりなモニタがNECから発売されました。まるで，X680x0シリーズのために作ったのではないの？　といえるようなこのモニタ。正直，CZモニタを愛用してきた人が，買い換えるのにちょうどいいモニタです。値段も79,800円で，コストパフォーマンスも最高です。

表示機能ほか

対応周波数は水平15.5kHz～65kHz/垂直55Hz～90Hzまでです。X680x0で作れる，ほとんどの画面モードを網羅しています。これだけの領域があれば，ゲームからSX-WINDOWでのメガディスプレイまで十分です。インタレースも90Hzまで垂直同期を上げれば，ちらつきが少なくなりますから，その気になれば，かなり広い画面がとれます。SX-WINDOW用として私がCRTC.X（1994年7月号の付録ディスクに収録されています）を使って作った画面モードを，表にしておきます。突き詰めればもっといきますが，十分見やすいレベルでここまでいけました（このあたりの詳しいことは1994年6，9月号を参照してください）。

ドットピッチは0.28mm。最近では当たり前のサイズです。画面のドット数を多くすると，その分，ドットピッチが小さいほうが文字がくっきり映ります。これだけ小さいサイズであれば，X680x0では十分でしょう。テレビ画面は，どちらかというとドットピッチが大きいほうが綺麗に映るものですが，映りはさほど悪くありません。多少白っぽくなりますが，これは小さなドットピッチの弊害でしょう。

スピーカーは，7.7cm丸型1.5W出力のものが左右1個ずつ。きちんとステレオで鳴らすことができます。ただ，CZモニタのようにモニタケーブルだけで，パソコンの音声を鳴らすことはできません。別系統にRGB音声入力端子があるので，そこへX680x0の音声出力端子をつなげます。この方法のほうが，音声出力に本体のノイズが乗りにくいですし，場合によっては，SC-55などのMIDI楽器を通しての接続もできるので便利かもしれません。

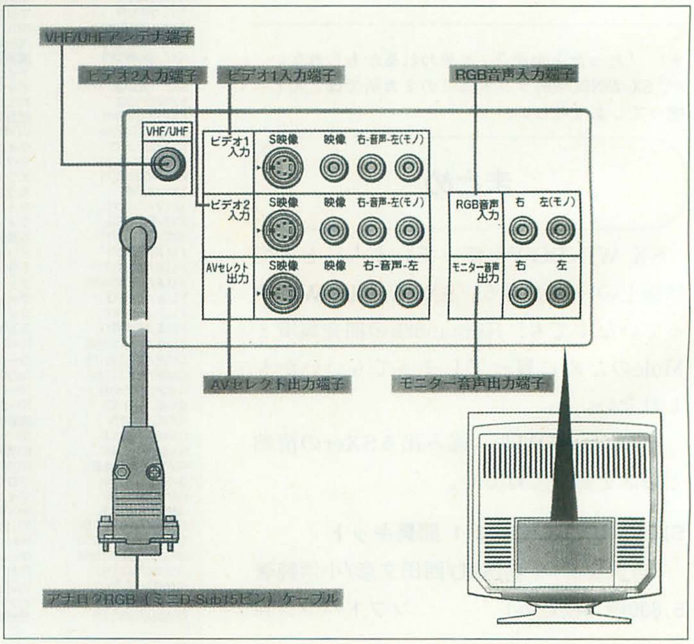
入力端子をまとめると，RGB用として，15pinシェルタイプが1系統。俗にいうDOS/Vマシン用のRGB端子ですが，変換アダプタが標準添付されているので，X680x0にも問題なく接続可能です（Macintosh用変換アダプタも付属）。ただ，変換アダプタを使うと通常のケーブルを接続するときよりも本体の奥行きを必要とするので注意が必要です。また，音声用として，先ほど説明したRGB音声入力が1系統。ほかには，VHF/UHF用アンテナ端子，ビデオ入力が2系統。出力端子は，AVセレクト出力が1系統。モニタ音声出力が1系統，正面にステレオヘッドホン端子が1系統あります（図）。

RGBの映像入力レベルは1Vp-p/75Ωです。X680x0の映像出力レベルは0.7Vp-p/75Ωですが，特に問題ないでしょう。いままでのPC-98用モニタはX680x0を接続すると暗くなることが多かったのですが，このモニタは自動的に色の濃さ，色合いをきちんと調整してくるので大丈夫です。これは映像コントロールウィンドウを本体ボタンやリモコンで呼び出せば手動での調整も可能です（写真1）。



PC-TM151　79,800円（税別）　NEC☎03(3452)8000

図　背面の入出力端子



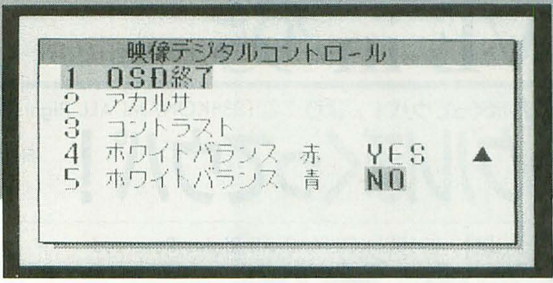


写真1 映像コントロールウィンドウ

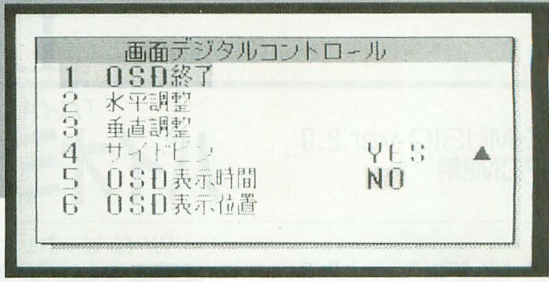


写真2 画面コントロールウィンドウ

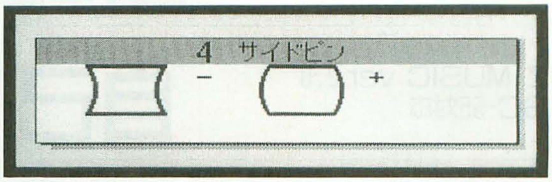


写真3 サイドピンの設定画面

また，映像コントロールウィンドウ以外にも画面コントロールウィンドウが呼び出せ，水平/垂直のサイズや位置，サイドピンの調節などができます(写真2)。サイドピンというのは，画面の左右が樽型に丸くなるか否かという設定です(写真3)。X680x0の画面モードは，設定しないと歪んでしまいますが，画面コントロールウィンドウで設定すれば問題はありません。ただし，X680x0の画面モードは水平のブランキング時間が妙に長いため，普通のモニタでは横いっぱいに広げることができません。限界まで広げすぎると，横エッジがふにゃふにゃになってしまうのです。したがって，通常の画面モードを利用すると，多少狭くなってしまいます（特殊な常駐ソフトを作れば問題ないのですが……)。

あと，画面サイズは型番からわかるとおり，15インチです。モニタの大きさは37cm（幅）×41cm（高さ）×43.1cm（奥行き)，重さは14.9kgです。

画面設定のコツ

このモニタは最近のマルチシンクモニタの例にもれず，同期周波数に合わせて，画面の設定をメモリすることができます。

X680x0の標準のIOCSの画面モードは2つあります。ひとつは15kHzのモードと，もうひとつは31kHzのモードです。24kHzもありますが，これが使われることは少ないでしょう。設定の例としては，SX-WINDOW用にノンインタレースとインタレースでの最大サイズを定義すると，あわせてX680x0だけで4つ，24kHzを利用するなら5つ必要です。編集室で試したところ5つの記憶が限界のようで，これ以上設定しようとすると，先に設定したものから忘れてしまいました。また，メモリするかどうかを自分で選択できず，画面上に画面コントロールウィンドウを出した瞬間からメモリされてしまうので，そのときだけ合わせるということができないのが残念です。画面コントロールウィンドウの操作は，比較的使いやすいのですが，いちいち調整するのはやっぱり大変です（ちなみに映像コントロールは各入力につき1種類ずつ設定できます)。

したがって，とりあえずSX-WINDOWの画面モードなどを設定して，最後に標準の画面モードを登録するのがベストでしょう。こうなると問題は，X680x0のゲームなどで，好き勝手に画面モードを作っているときでしょうか。メモリが足りなくなることがあるので注意が必要なのです。

それでもほかのパソコンで，メジャーなものはROMに入っているようで，ディスプレイセレクタなどを利用して，PC-9801と切り替えたりしても問題はないでしょう。

なお，わかる人だけしか意味がわからないかもしれませんが，RGB入力はシンクオングリーンやC-SYNCでも可能です。これに自動輝度設定機能があるとなると……。

まとめ

映りなどは，最近のモニタに比べると，あまりいいほうではありません。調整もあまり細かくできないので，多少，エッジが湾曲することもあります。細かい問題はいろいろありますが，1世代前のCZモニタに比べたらずっと綺麗です。

値段もかなり安いし，いろいろなハードをつなげることができます。複数のパソコンを1台のモニタで共有すると，なかなかすべてを綺麗に映すことはできませんが，だいたい丁度いい感じで設定されていると思います。ここまで，そのままCZモニタとして置き換えることができ，さらに機能が増えているものはほかにありませんからね。S端子など，CZモニタにはとうとうつかなかったものがあるのも魅力です。

あとは，色が黒で，X680x0用のテレビコントロール端子さえあれば，完璧だったんですけどね（笑)。

表 PC-TM151メガディスプレイ計画

インタレースモード	インタレース	インタレースモード	ノンインタレース
ドットクロック	34.776MHz(69/2)	ドットクロック	34.776MHz(69/2)
水平表示	952dot(27.375μs)	水平表示	864dot(24.845μs)
水平パルス	72dot(2.070μs)	水平パルス	56dot(1.610μs)
水平バックポーチ	104dot(2.991μs)	水平バックポーチ	56dot(1.610μs)
水平フロントポーチ	16dot(0.460μs)	水平フロントポーチ	32dot(0.920μs)
水平同期	32.896μs 30.399kHz	水平同期	28.986μs 34.500kHz
垂直表示	714line(11.744ns)	垂直表示	648line(18.783ns)
垂直パルス	2line(0.066ns)	垂直パルス	2line(0.058ns)
垂直バックポーチ	14line(0.461ns)	垂直バックポーチ	14line(0.406ns)
垂直フロントポーチ	0line(0.000ns)	垂直フロントポーチ	0line(0.000ns)
垂直同期	12.287ms 81.388Hz	垂直同期	19.246ms 51.958Hz
Reg0 =	142(8e)	Reg0 =	125(7d)
Reg1 =	8(8)	Reg1 =	6(6)
Reg2 =	17(11)	Reg2 =	9(9)
Reg3 =	136(88)	Reg3 =	117(75)
Reg4 =	372(174)	Reg4 =	663(297)
Reg5 =	1(1)	Reg5 =	1(1)
Reg6 =	15(f)	Reg6 =	15(f)
Reg7 =	372(174)	Reg7 =	663(297)
Reg20 =	26(1a)	Reg20 =	22(16)
HRL =	0	HRL =	0

Oh!X LIVE in '95

「バイオミラクルぼくってウパ！」より　©1988KONAMI ALL Rights Reserved

Z-MUSIC ver.2.0
PCM8用

バイオミラクルぼくってウパ！

Sahara Masaharu 佐原 政治

「ツインビーヤッホー！」より　©1995KONAMI All Rights Reserved

Z-MUSIC ver.2.0
SC-55対応

君に会うために……

Morikami Akihito 森上 晶仁

「闇の血族」より　©SYSTEM SACOM

Z-MUSIC ver.2.0
SC-55対応

自己紹介のテーマ

Yamazaki Mikio 山崎 幹生

Z-MUSIC ver.2.0
SC-88対応

TIME STREAM

Matsuo Naoki 松尾 直樹

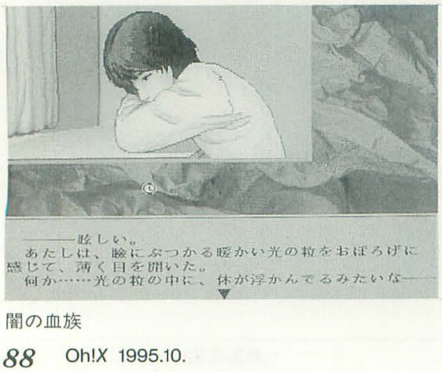
音楽の秋。今月はゲームミュージックを中心に4曲お送りしましょう。昔懐かしいファミコンサウンドから最新の「ツインビーヤッホー！」，マサ斎藤氏の名作「闇の血族」，さらにSC-88用のデータも掲載します。

合言葉はウパっー！

まずは1曲め，内蔵音源の元気なサンバ調のゲームミュージックからお届けします。コナミのファミコンゲーム「バイオミラクルぼくってウパ！」のメインテーマです。

原曲は，いまとなってはビンテージサウンドの地位を確立しつつあるファミコンPSGによるものです。これを佐原君はADPCMラテンパーカッションを加え，ティンバーパートはオリジナルサウンドのイメージを踏襲したFM音源音色でリニューアルしてくれました。

聞いてみるとそんなにパート数は多くないのにとても深みと広がりのある演奏に聞こえますね。これは古くからの本コーナー読者にはお馴染み，ディレイとディチューンテクによるものです。たとえば，この曲



闇の血族

ではチャンネル1～3の3チャンネルをメロディに振り分け，これを左，中央，右のパンポットにスプリットしています(3チャンネルとは贅沢ですね)。それぞれを微妙にピッチを変えて(ディチューンテク)，いちばん音量を小さくしたパートを他パートよりもやや遅らせて(ディレイテク)演奏させています。

ただでさえ少ないチャンネルを余計に使ってしまうのが難点ですが，エフェクタのないFM音源において分厚い響きを作り出すための必須テクです。

さて，演奏にはZ-MUSIC ver.2.0のほかにPCM8.Xと「Z-MUSICシステムver2.0」(ソフトバンク刊)に収録されているADPCM音ファイルが必要です。これらをセットアップした環境でリスト1のAD PCMコンフィギュレーションファイル＊1を入力し，

A>ZPCNV filename.CNF

としてZPDを作成してください。次にリスト2の曲データ，ZMSファイルを行番号抜きで打ち込み，

A>ZMUSIC -Q filename.ZMS

を実行してください。このときの出力がリスト3と一致していれば正しく入力ができています。間違っていたら間違っているトラックをチェックしてみましょう。

入力したZMSに間違いがないことを確認できたら，

A>ZP filename

で演奏できます。

＊1AD PCMコンフィギュレーションは編集室で多少修正を施しました。

合言葉はヤッホー！

お次はSC-55(GS音源)用の曲です。「ツインビーヤッホー！」よりステージ6のテーマ「君に会うために……」をお届けしましょう。「ツインビーヤッホー！」の曲は一度聞いたら耳から離れないメロディアスソングが多くて，そう，ゲームミュージックの原点ともいえる魅力を持っています。ステージ1～3の曲とかの途中の盛り上がりでのお約束の転調，「お約束」とわかっていながらも聞いてて思わず熱くなってしまいます。アレンジバージョンCDのほうもいいのでぜひ一度聞いてみてください。できたらコピーに挑戦するのもいいかも。

さて，森上君の選んだこの曲はオーケストラポップス的な構成の，実に音楽性の高い曲です。ひとつの楽器がひとつのパートを演奏するのに留まらず，曲の進行に応じてバックに回ったりリードに回ったりしています。リリカルでありながらも同時に大

変技巧的なのです。こういうオリジナル曲ができれば，一人前なんでしょうが難しいですよねぇ。

曲の出来は限りなく満点に近く，安心して聞けます。ただしSC-55の発音の関係からか中盤の盛り上がりが原曲よりややおとなしめです。

演奏にはSC-55が必要です。SC-55mkII/SC-88ではややニュアンスが異なって演奏されます。

懐かしの「ちょっとため息」

次はシステムサコムの名作アドベンチャー「闇の血族」より「自己紹介」のテーマです。

この通称「闇ケツ」，前編は殺人ミステリー，後編はサイキックアクションという一風変わった作品でした。このゲームの主人公魅由の口癖「はふ，ちょっとため息」はX68000ユーザーの間で「闇ケツ」語として広まり，密やかなブームにもなりました。この「自己紹介」はそんな思い出深いこのゲームの冒頭曲です。ゲームでは迫力のオープニング終了後，この曲をバックに主人公魅由が(AD PCMで)かわゆく(!?)自己紹介をしだします。

MT-32で演奏されていた原曲を同系のGS音源音色にグレードアップした感じのできあがりで，実にそつなくまとまっています。

AメロのSaw Waveはオクターブユニゾンでジュージューしたシンセ臭いハーモニーを作り出しています。これがなんともLA音源(MT-32)ぽくていいですね。

Aメロ→Bメロ→Cメロの展開がいかにも作曲者故斎藤学氏らしいです。

演奏にはSC-55(GS音源)が必要です。編集室でSC-55mkII/SC-88での正常な演奏を確認しました。

時の流れに身をまかせ

Oh!X LIVEではすっかり常連顔になりましたお馴染み松尾君の作品です。今回はお得意のフュージョンネタをSC-88で送ってきてくれました。カシオペア「TIME STREAM」です。

カシオペアの曲は野呂一生が弾くギターがリードパートとなる場合が多いですよね。局面局面で微妙にニュアンスが変化するこの野呂ギターをいかにリアルにコピーするかが，DTMフリークにとっては最大の難問であり「やりがい」でもあるのです。松尾君もこれには相当苦労したようで，このパート(トラック1,9)は拡張ピッチモジュレーション，拡張ARCC，ベロシティシーケンスなどの複数のZ-MUSICの特殊機能を駆使して野呂ギター再現を試みています。

それと，このデータはバックパートも念入りに作られています。速いテンポの曲を聞くときはそれほど目立たない演奏の「緩急」も，スローテンポの曲では非常に重要なファクターになります。聞き手を引きつけるためには，スローテンポの曲をシーケンスするときは特に(ハーモニー感ももちろんですが)，演奏に感情/情緒を盛り込むことが必要不可欠になってくるでしょう。そうでないとただの遅い曲という印象を聞き手に与えてしまいかねません。そういう意味でも今回のデータは大変完成度が高いものになっています。特にリズムとピアノ，



CASIOPEA

ここまで気合入れる必要があるのかと思うほどリアルに再現されています。

リストはチョット長めですが気合を入れて打ち込んでください。演奏にはSC-88が必要です。

リストでちょっと目を引いたのが75行からのエフェクトの設定部分です。MMLの@Eコマンドはそれぞれリバーブ，コーラスのかかり具合を設定するものですが，その後ろのy94,nはいったいなにを行っているのでしょうか。そう，ディレイエフェクトの設定です。SC-88ではSC-55のエフェクタセットに加えて新たにディレイが追加されました。これがコントロール94に割り当てられたのでy94,nとしてこのディレイエフェクトのかかり具合を設定しているわけです。ちなみにコントロール番号91〜95の5つのコントロールは汎用エフェクトコントロールとしてMIDI規格で定められています。SC-88では91番(REVERB)と93番(CHORUS)，94番(DELAY)が使われているのです。あと残りの2つもそのうちSC-88などの後継で機能実装されるかもしれませんね。個人的には95番にディストーションかなんかがほしいところですね。 (Z.N.)

リスト1 バイオミラクルぼくってウパッ!

```
================  UPA.ZMS  ==================
  1: .comment  バイオミラクルぼくってウパ!メインテーマ  MASA1995
  2:
  3: /-----------------------------------------------------------
  4: / TRACK SETUP
  5:
  6: (i)
  7: (b0)
  8:
  9: (m1,3000)(aFm1,1)
 10: (m2,3500)(aFm2,2)
 11: (m3,4000)(aFm3,3)
 12: (m4,3500)(aFm4,4)
 13: (m5,4500)(aFm5,5)
 14: (m6,5000)(aFm6,6)
 15: (m7,2500)(aFm7,7)
 16: (m8,3000)(aFm8,8)
 17: (m9,3000)(aAdpcm,9)
 18: (m10,3500)(aAdpcm,10)
 19: (m11,2000)(aAdpcm,11)
 20: (m12,2000)(aAdpcm,12)
 21:
 22: / ADPCM DATA SET
 23:
 24: .ADPCM_BLOCK_DATA = upa.ZPD
 25: /-----------------------------------------------------------
 26: / OPM DATA SET
 27:
 28: /  AR 1DR 2DR  RR 1DL  TL  RS MUL DT1 DT2 AME   マリンバ
 29: (@1,         20, 18,  4,  2,  2, 51,  2,  3,  3,  0,  0
 30:     31,  0, 10,  8,  0,  0,  2,  0,  3,  0,  0
 31:     22, 20,  4,  2,  4, 26,  2,  3,  7,  0,  0
 32:     31, 15, 13,  8,  2,  0,  2,  0,  7,  0,  0
 33: /  AL  FB  OM
 34:      4,  7, 15)
 35: /
 36: /  AR D1R D2R  RR D1L  TL  RS MUL DT1 DT2 AME   E Bass
 37: (@2,         27, 14,  0,  6,  3, 28,  0,  8,  0,  0,  0
 38:     31, 12,  0,  6,  3, 47,  0,  2,  0,  0,  0
 39:     31, 19,  0,  6,  5,  5,  0,  0,  0,  0,  0
 40:     31,  6,  3,  7, 14,  4,  0,  0,  0,  0,  0
 41: /* AL  FB  OM
 42:      3,  7, 15)
 43: /
 44: /  AR D1R D2R  RR D1L  TL  RS MUL DT1 DT2 AME   MAIN1
 45: (@3,         28, 12,  6,  7,  0, 32,  0,  6,  3,  0,  0
 46:     28,  5,  6,  7,  3,  0,  0,  4,  3,  0,  0
 47:     28, 12,  6,  7,  0, 32,  0,  4,  7,  0,  0
 48:     28,  5,  6,  8,  3,  0,  0,  4,  7,  0,  0
 49: /* AL  FB  OM
 50:      4,  7, 15)
 51: /
```

```
/  AR D1R D2R  RR D1L  TL  RS MUL DT1 DT2 AME   MAIN2
(@4,        31, 12,  6,  0,  5, 30,  0,  2,  3,  0,  0
    31,  8,  6,  8,  5,  0,  0,  3,  3,  0,  0
    31, 12,  6,  0,  5, 30,  0,  2,  7,  0,  0
    31,  8,  6,  8,  5,  0,  0,  4,  7,  0,  0
/* AL  FB  OM
     4,  7, 15)
/
/  AR D1R D2R  RR D1L  TL  RS MUL DT1 DT2 AME   MAIN2
(@5,        31,  2,  3,  2,  5, 26,  1,  7,  0,  0,  0
    31,  2,  3,  2,  7, 28,  1,  6,  0,  0,  0
    31, 10,  3,  2, 10, 35,  1,  2,  0,  0,  0
    31,  7,  6,  5,  6,  0,  2,  1,  0,  0,  0
/* AL  FB  OM
     2,  7, 15)
/
/  AR D1R D2R  RR D1L  TL  RS MUL DT1 DT2 AME   Guitar
(@6,        31,  9,  0,  6, 15, 39,  1,  8,  5,  0,  0
    31, 10,  0,  9, 15, 29,  1,  3,  3,  0,  0
    28,  6,  0,  8, 15, 34,  0,  3,  5,  0,  0
    31, 18,  8,  8,  0,  1,  1,  1,  3,  0,  0
/* AL  FB  OM
     0,  7, 15)
/
/  AR D1R D2R  RR D1L  TL  RS MUL DT1 DT2 AME   HARP
(@7,        31, 10,  0,  3,  5, 54,  0, 15,  3,  0,  0
    31, 10,  0,  5,  5, 29,  0,  0,  7,  0,  0
    31, 10,  0,  5,  1, 17,  0,  0,  1,  0,  0
    31,  4,  5, 12, 10,  0,  0,  2,  7,  0,  0
/* AL  FB  OM
     3,  7, 15)
/*------------------------------------------------------------
(T1) /Main Dt+
     t135
     v13o4@k+5
     [do]
     o4@s5@m30@h36l16
     @3p2_2ggrgrgrgg4r4l8
     |:e4c4|d4g4r>a4<c>b16r16b16r16<c16d.:|
     g4r16d16g16b16<c16r16c16>b16r16<c16r16>g2^16
     @4p2^2|:rga<cdc>b16g.g32g+32a16e|e-16d16r16>a2^16
     rbbb-bb-b16<d.c16r16c16r16c16d16r16e2^16:|
     d16e16r16>a2^16l16bababr<cdr>gb<degag<crcr>a<crcr4
     ^4@5p2>c8d8|:eerere-ee-er>g+*3a*21<c8e8ffrfrerf>g+*3a*45r8a8|
     a*3b*9b-rbrb-bb-b8<c8d8>b8b*3<c*9>grg8grag8<<e-ee-dc>g:|
     ddrdrcdc @3p2_4>d<d>e<e>f<f>f+<f+>
     [loop]
/*------------------------------------------------------------
(T2) /Main Dt-
     v13o4@k-5
     [do]
     o4@s5@m30@h36l16
     @3p1_2ggrgrgrgg4r4l8
     |:e4c4|d4g4r>a4<c>b16r16b16r16<c16d.:|
     g4r16d16g16b16<c16r16c16>b16r16<c16r16>g2^16
     @4p1^2|:rga<cdc>b16g.g32g+32a16e|e-16d16r16>a2^16
     rbbb-bb-b16<d.c16r16c16r16c16d16r16e2^16:|
     d16e16r16>a2^16l16bababr<cdr>gb<degag<crcr>a<crcr4
     ^4@5p1>c8d8|:eerere-ee-er>g+*3a*21<c8e8ffrfrerf>g+*3a*45r8a8|
     a*3b*9b-rbrb-bb-b8<c8d8>b8b*3<c*9>grg8grag8<<e-ee-dc>g:|
     ddrdrcdc @3p1_4>d<d>e<e>f<f>f+<f+>
     [loop]
/*------------------------------------------------------------
(T3) /Main Echo
     v10o4r16.
     [do]
     o4@s5@m30@h36l16
     @3_2ggrgrgrgg4r4l8
     |:e4c4|d4g4r>a4<c>b16r16b16r16<c16d.:|
     g4r16d16g16b16<c16r16c16>b16r16<c16r16>g2^16
     @4^2|:rga<cdc>b16g.g32g+32a16e|e-16d16r16>a2^16
     rbbb-bb-b16<d.c16r16c16r16c16d16r16e2^16:|
     d16e16r16>a2^16l16bababr<cdr>gb<degag<crcr>a<crcr4
     ^4@5>c8d8|:eerere-ee-er>g+*3a*21<c8e8ffrfrerf>g+*3a*45r8a8|
     a*3b*9b-rbrb-bb-b8<c8d8>b8b*3<c*9>grg8grag8<<e-ee-dc>g:|
     ddrdrcdc @3_4>d<d>e<e>f<f>f+<f+>
     [loop]
/*------------------------------------------------------------
(t4) /Bass
     [do]
     @2v15o3l16
     ggrgrgrggr8.>gg<g>g<
     l8@q2c.>g.<c>b.g.ba.e.al16grgrgl8b.<
     |:c.>g.<c>b.g.ba.e.ag.a.b<:|
     d.>g.<d>g.<d.>g<c.>g.<c>g.a.b<
     c.>g.<c>b.g.ba.e.ae.a.eg.d.gd.g.<dl16crcrgcrcl8r4c>b
     l16|:a8.araraa8r8a8g8f8.frfrff8r8f8a8
     g8.grgrg|g8a8b8<d8c8.crcrcc8r8c8>b8:|
     <d8e8f8f+8
     [loop]
/*------------------------------------------------------------
(t5) /Sub 1
     [do]
     @k+3
     @s5@m30@h36
     v11o4@3p1o3l16
     bbrbrbrbb4r4
     @1p1o6p1|:e8g8ce8ddg8>b< c8d8c8e8>a<c8>bb<d8>b<c8d8:|
     e8g8ce8d8g8dd8g8c8e8>a<c8>bb<d8>gb8<d8
     >b8<d8>gb8bb<d8>gb8<d8c8e8>g<c8cg>gg<a>aa<b>b<
     e8g8ce8ddg8>b<d8g8c8e8>a<c8cce8>a<c8e8>b8<d8>gb8bb<d8>gb8<d8grgrfgrgr2
     o3@6p1v14|:ea<ceaea<c>a<ceaec>ae
     c>a<fcafa<cfcfafc>af d>b<gdbgb<d|gdgbgd>bg ecge<c>ce8r8
     >b<cbb-ae:|_2>d<d>e<e>f<f>f+<f+>
     [loop]
/*------------------------------------------------------------
(t6) /Sub 2
     [do]@k-3
     @s5@m30@h36p2
     v11o4@3o3l16
     bbrbrbrbb4r4
     @1o6p2|:c8c8>g<c8>gbb8g g8b8a8<c8>ea8egb8gg8b8<:|
     c8c8>g<c8>g8b8bg8g8a8<c8>ea8egb8dg8b8
     g8b8dg8gdg8dg8b8g8g8eg8gg>gg<a>aa<b>b<
     <c8c8>g<c8>bgb8gb8b8a8<c8>ea8ea<c8>ea8<c8>g8b8dg8dgb8dg8b8<ererderer2
     o3@6p2v11r16.|:ea<ceaea<c>a<ceaec>ae
     c>a<fcafa<cfcfafc>af d>b<gdbgb<d|gdgbgd>bg ecge<c>ce8r8
     >b<cbb-ae:|_2>d<d>e<e>f<f>f+32
     [loop]
/*------------------------------------------------------------
(T7) /Back 1
     [do]
     r1@m10@s2@k+5
     r1r1r1r2@7v12l16o7gd>bgd>bgd
     r1r1r1r2g>gg<a>aa<b>b<
     r1r1r1r2.
     c8d8|:eerere-ee-er>a8<c8e8ffrfrerf>a4r8a8|
     bb-rbrb-bb-b8<c8d8>b8<c>grg8grag8<<e-ee-dc>g:|
     ddrdrcdc >d<d>e<e>f<f>f+<f+>
     [loop]
/*------------------------------------------------------------
(T8) /Back 2
     r16.
     [do]
     r1@m10@s2@k-5
     r1r1r1r2@7v10l16o7gd>bgd>bgd
     r1r1r1r2g>gg<a>aa<b>b<
     r1r1r1r2.
     c8d8|:eerere-ee-er>a8<c8e8ffrfrerf>a4r8a8|
     bb-rbrb-bb-b8<c8d8>b8<c>grg8grag8<<e-ee-dc>g:|
     ddrdrcdc >d<d>e<e>f<f>f+<f+>
     [loop]
/*------------------------------------------------------------
(t9) /Conga
     @1
     [do]
     o4v3l16ffrf rfrff2
     o4|:12e8f8e8feefeee8f8:|
     |:14rerefrre:|
     [loop]
/*------------------------------------------------------------
(t10) /Agogo
     @1v4
     [do]
     o5l16ccccrcrcrcccdccd
     |:12crcrddrcrcrcdrdr:|
     |:14dcdcddcd:|
     [loop]
/*------------------------------------------------------------
(t11) /Bass & Snare & Tom
     l16v8o1[do]@r1
     ddcdcdrccddc<<ccde>>
     o1|:11crcrdr<<ce>>cddcdrcd:|drdrcdrdcc<<ccddee>>
     |:8c|rrrdr<<de>>:|ddd<<ddee>>
     |:6c|rrrdr<<de>>:|dddddd[loop]
/*------------------------------------------------------------
(t12) /Hihat
     [do]
     l16v7o2
     ccd8cd8.c|:6c:|d
     |:24ccd8cccc:|
     |:14ccd8crcc:|[loop]
/*------------------------------------------------------------
(p)
```

リスト2　バイオミラクルぼくってウパッ！用コンフィグファイル

```
================  UPA.CNF  ==================
.adpcm_bank 1
.o1c=  fck.PCM
/.o1d=  tr909sd.PCM
.o1d=  housesn.pcm
.o3c=  atkt3.PCM
.o3d=  atkt2.PCM
.o3e=  atkt1.PCM
.o2c=  tr808Hc.PCM
.o2d=  tr808ho.PCM
.o4e=  conga1.pcm
.o4f=  conga2.pcm
.o5c=  agohi.pcm
.o5d=  agolo.pcm
```

リスト3　バイオミラクルぼくってウパッ！用カウンタ表示

```
 1:00000000 00000F00   2:00000000 00000F00   3:00000012 00000F00   4:00000000 00000F00
 5:00000000 00000F00   6:00000000 00000F00   7:00000000 00000F00   8:00000012 00000F00
 9:00000000 00000F00  10:00000000 00000F00  11:00000000 00000F00  12:00000000 00000F00
```


▶私の行く本屋ではOh!Xは店頭にありません。行くと店のおばさんが，直接くれます。私のため，１冊のみ取り寄せてくれているようです。　熊倉　隆(23)新潟県

リスト4 ツインビーヤッホー！

```
===============  TWYH_ST6.ZMS  ==================
/              「ツインビー ヤッホー！」より
/              「君に会うために…(STAGE 6)」
/                         (C)1995 KONAMI
/      音楽  コナミ矩形波倶楽部(エンゾニック前田)
/                採譜・打ち込み  カシュアンヌ
/ ○制作日誌
/ '95.7.12夜        採譜(メロディ、ベース、コード)
/         12~15     細部を採譜しつつ MML を作成
/         16         最終調整
.comment SC-55: ツインビー ヤッホー! "君に会うために (STAGE 6)" (C)19
95 KONAMI
/=================== Z-MUSIC Reset
(i)(b1)(z192)
/=================== Allocation & Assignment
(m1,2000)(aMIDI1,1)          / Bass
(m2,2000)(aMIDI2,2)          / Synth 1
(m3,3000)(aMIDI3,3)          / Strings
(m4,2000)(aMIDI4,4)          / Horn
(m5,2000)(aMIDI5,5)          / Brass
(m6,2000)(aMIDI6,6)          / Synth 2
(m7,2000)(aMIDI7,7)          / Electric piano & Pad
(m8,2000)(aMIDI8,8)          / Flute & Oboe & Fagott
(m9,2000)(aMIDI9,9)          / Orchestra hit & Bell
(m10,2000)(aMIDI11,10)       / Timpani
(m11,2000)(aMIDI10,11)       / Bass drum
(m12,2000)(aMIDI10,12)       / Snare drum & Tambourine
(m13,2000)(aMIDI10,13)       / Cymbal
(m14,2000)(aMIDI12,14)       / Electric tomtom
/=================== MIDI Module Reset
(t1)
 t60         / Ex が確実に送信出来る様、遅めに設定
 @i$41,$10,$42      / model ID:GSstandard
 x$40,0,$7f, 0      / system reset
 r8          / ← リセット後のウェイトは長めにする
 x$40,1,$10, 4,1,2,2,2,2,2,2,1,2,1,1,0,0,0,0 r*1
   / voice reserve
   / (Part10,1,2,…,9,11,…,16 の順であることに注意)
   / ☆★がぁ~!パーシャル全然足らん!★☆
 x$40,1,$30, 4,4,0,0,127,0,55 r*1
   / reverb macro:4 (Hall2)
   / reverb character:4 , reverb pre-lpf:0
   / reverb level:0     , reverb time:127
   / reverb feedback:0  , send level to chorus:55
 x$40,$01,$38, 4,0,127 r*1
   / chorus macro:4 (FeedbackChorus)
   / chorus pre-lpf:0   , chorus level:127
 x$40,1,0, 0,$54,$77,$69,$6e,$42,$65,$65,0,$59,$6f,$2d,$68,
$6f,$21,0 r*1
   / " TwinBee Yo-ho! ": SC-55の[ALL]を押すと表示される
 x$40,$12,$14, 0 r*1        / Part2 single assign mode
 x$40,$13,$14, 0 r*1        / Part3         〃
 x$40,$14,$14, 0 r*1        / Part4         〃
 x$40,$15,$14, 0 r*1        / Part5         〃
 x$40,$16,$14, 0 r*1        / Part6         〃
 x$40,$17,$14, 0 r*1        / Part7         〃
 x$40,$18,$14, 0 r*1        / Part8         〃
 x$40,$19,$14, 0 r*1        / Part9         〃
 x$40,$10,$14, 2 r*1        / Part10 full-multi assign mode
 x$40,$1b,$15, 2 r*1        / Part12 for drum part 2
 @i$41,$10,$45              / model ID:SC-55
 x$10,0,0, $54,$77,$69,$6e,$42,$65,$65,0,$59,$6f,$2d,$68,$6
f,$21,0,$54,$6f,0,$4d,$65,$65,$74,0,$59,$6f,$75,0,$28,$53,$74,$3
6,$29 r*1
   / .sc55_print "TwinBee Yo-ho! To Meet You (St6)"
/=================== Tempo, ID, Tie Mode, Bend Range
(t1) t155            @i$41,$10,$42 "1         @g12
(t2,3,4,5,6,7,8,9) @i$41,$10,$42 "1 r8^*15 @g12
(t10)                @i$41,$10,$42 "1 r8^*15 @g12
(t11,14)             @i$41,$10,$42    r8^*15 @g12
(t12,13)                              r8^*15
/=================== Controll Change
(t1) @33 @v124 @p64 @e0,6 y$7e,0   / Bs.
(t2) i8 @49 @v116 @p56 @e60,10             / Syn.1
(t3) i127 @50 @v122 @p27 @e60,0            / Str.
 / release time:+3
 @x$b2, $63,1,$62,$66,6,67
 x$40,$13,$1a, 77,45         / velocity に対する音量変化を変更
(t4) i8 @64 @v116 @p88 @e50,16             / Hr.
 / decay time:+40
 @x$b3, $63,1,$62,$64,6,104
(t5) i8 @62 @v116 @p98 @e50,16             / Br.
 / attack time:-2, decay time:+40
 @x$b4, $63,1,$62,$63,6,62, $62,$64,6,104
(t6)      @v96       @e,0                  / Syn.2
(t7)      @v96       @e,0                  / EP & Pad
(t8) @71 @v96 @p42 @e75,0          / Fl. & Ob. & Fg.
 / vibrato (rate:+7, depth:+15, delay:+2)
 @x$b7, $63,1,$62,8,6,71, $62,9,6,79, $62,$a,6,66
 / release time:+6
 @x$62,$66,6,70
(t9)      @v86                             / Ohit. & Bl.
(t10) @48 @v124 @p64 @e60,20 y$7e,0        / Timp.
 / cutoff freq:-7, resonance:+8
 @x$ba, $63,1,$62,$20,6,56, $62,$21,6,72
(t11) @1 @v116 @p64 @e70,0          / Perc.1
 @x$b9, $63,1,$62,$64,6,68
 @x      $62,38,$63,$18,6,65,$63,$1d,6,80           / SD1
 @x      $62,40,$63,$18,6,65,$63,$1d,6,127          / SD2
 @x      $62,42,$63,$18,6,63
/ Hhc
 @x      $62,46,$63,$18,6,63
/ Hho
 @x      $62,49,$63,$18,6,63,$63,$1c,6,92,$63,$1d,6,127
/ Crsh1
 @x      $62,52,$63,$18,6,60,$63,$1d,6,127          / Chns
 @x      $62,55,$63,$18,6,63,$63,$1d,6,127          / Splsh
 @x      $62,57,$63,$18,6,62,$63,$1c,6,36,$63,$1d,6,127
/ Crsh2
 x$41,0,0, $54,$77,$69,$6e,$42,$65,$65,0,$44,$72,$2e,$31 r*
1
 x$41,3,36, 7,0,8,0,9 r*1
 x$41,3,54, 10
(t14) @25 @v110 @p64 @e70,20               / Perc.2
 @x$bb, $62,41,$63,$18,6,62,$63,$1c,6,  1,$63,$1d,6,127
/ Elt2
 @x  $62,45,$63,$18,6,62,$63,$1c,6, 64,$63,$1d,6,127 / Emt2
 @x  $62,48,$63,$18,6,62,$63,$1c,6,127,$63,$1d,6,127 / Eht2
 x$41,$10,0, $54,$77,$69,$6e,$42,$65,$65,0,$44,$72,$2e,$32
r*2
/ ARCC、Expression、調号設定
(t1,2,3,4,5,6,7,8) m,1 s,3 @c11,100,100 y11,100 r*2^4
(t9,10)            m,1 s,3 @c11,100,100 y11,100 r*2^4
(t11,14)           @r1                  y11,100 r   4
(t12,13)           @r1                          r*2^4
(t1,2,3,4,5,6,7,8) [k.sign +c,+f,+g]       / イ長調
(t9,10)            [k.sign +c,+f,+g]       /   〃
/=================== Bass
(t1)        u118 L8 o1 r4 |:16@q3e<e:|
   [do] |:32>a:| |:16g:| |:16e:|
   |:16<d:| |:8c:| |:8c!:| |:8>b:| eee<edddd
   cccc>e+e+e+e+ |:8f:| |:7<d:|c |:7>b:|a |:8g!:| |:8e:|
   |:4aq6a^@q3a:| |:gq6g^@q3g:| |:fq6f^@q3f:|
   L4 |:8f!:| g!g!g!g! eeee
/=================== Synth 1
(t2)        r4 o4 u88 L4 @q2a2.{aaa} b2.{bbb}
   <c!2c!{c!c!c!} u+4d2u+4g!2
   [do] u88 L2 r>e a<c4>a4 b^8f8q3b8<q5c8 @q2d+1
   r>f b<d+ fe >b4<u+4c4u+4d4u+4e4
   L8 u+4f2>u-20f2 u+10a4<u+6d4u+7f4u+8a4
   au-10gu-10e4u+20au-10gu-10du+10q8e u+10@q2b4.a^2
   u-22q5a4.@q2d^fa<u+4d u+5c4>b4a4g4 b2.<c4 >q7a1
   u+5@q2f2g4a4 <u+6d4.c>b<c>ba <q7e1& e1
   L4 >>u-45q7c2u+8q7eu+8a u+8fu+10b<u+10fu-9e8u-9d+8
   u+9eu-10>b<u+20bu-9a8u-9g8 u+9au-10e<u+20eu-9d8u-9c8
   >>q7u+10f!<f!edc!>b-af! @q2g!g!8g!8q7b2 gu+5b<u+5eu+5g
/=================== Strings
(t3)        r4 o4 u55 L1 @q3'd>f' 'e>g' 'f!>a' 'g!>b'
   [do] z96,-10,+4 L8 |:4<q4'ec''ec'r'ec':|
   |:8>'b<d+''b<d+'r'b<d+':| |:'b<d''b<d'r'b<d':|
   z96,-3,+3,+3,+3,-3,+3,+3 L16 @q1g!fg!abab<c
   z+3,-6,+3,+3,+3,-6,+3,+3 d>b<cdecde
   L8 u+3q7'fd2'u-15@q1'>af'u+6'd>a'u+6'fd'u+6'af'
   u-12'af4'u-2q7'fd4'u-2@q1'fd4'u-2q7'd>a4'
   u-2@q1'c2.>a''d>b'u+9'ec' u+9q7'be4.'&'ae^2'
   u-50'af4.''d>b^''fd''af''b<d'
   u+8@q1'<cf!4''b<d4'u-3'a<c4'u-3'gb4' 'b<d2.''<ce4'
   'a<c1' u-8'fd1' 'fd1' 'g!e1' 'ae2'&'ge2'
   u64 L4 q7'c>a2'u+8@q4'ec'u+8'ae'
   u+8'fd+'u+10'bf'<u+10'fd+'u-9@q3'ec8'u-9'd+>b8'
   u+9@q4'e>b'u-10'>bg'u+20'bg'u-9@q3'af8'u-9'ge8'
   u+9@q4'ae'u-10'ec'<u+20'ec'u-9@q3'd>b8'u-9'c>a8'
   z90,-4,-4,-4,+8,-4,-4,-4,+8,-4,-4,-4
   L16 |:@q1'f!>f!''e>e''d>d''c!>c!':|
   'e>e''d>d''c!>c!''>b->b-''d>d''c!>c!'>'b->b-''a>a'
   z+8,-4,-4,-4,+8,-4,-4,-4,+8,-4,-4,-4,+8,-4,-4,-4
   '<c!>c!''b->b-''a>a''g!>g!''b->b-''a>a''g!>g!''f!>f!'
   'a>a''g!>g!''f!>f!''e>e''f!>f!''e>e''d>d''c!>c!'
   z-8,+4,+4,+4,-6,+4,+4,+4,-6,+4,+4,+4,-6,+4,+4,+4
   >'g!>g!''a>a''b>b''c<c''a>a''b>b'<'c>c''d>d'
   '>b>b''c>c''d>d''e>e''c>c''d>d''e>e''f>f'
   z-6,+4,+4,+4,-6,+4,+4,+4,-6,+4,+4,+4,+4,+4,+4,+4
   'e>e''f>f''g>g''a>a''g>g''a>a''b>b''c<c'
   'b>b'<'c>c''d>d''e>e''f>f''g>g''a>a''b>b'
/=================== Horn
(t4)        r4 o4 u85 L4 q7f2.{fff} g2.{ggg} a2a{aaa} u+4b2<
u+4d2
   [do] u98 L2 r>q7c ea4e4 f^8d+8q5f8a8 @q8b1
   rq7d+ d+f bg! g!4u-3a4u-3b4<u-3c4
   u-3d>a fd c.>b4 <q7c!1
   f1 @q2f!q7(f!g16)&g4.. e+e+ f1
   >u+6@q2aq7(a<d16)&d4.. >@q2aq7(<df32)&f4...
   @q2g!1 aq7g
   u-16q7ea fr8u+5q3d+8e8f8 u+5q7gu+5(gb32)&b4...
   u-10e4c4<c4>b8a8
   L4 q7c!<c!c!>aa f!f!c! @q2dd8d8q7g!2 eu+5gu+5b<u+5e
/=================== Brass
(t5)        r4 o4 u88 L4 q7a2.{aaa} b2.{bbb} <c!2c!{c!c!c!}
   u+4d2u+4g!2
```

▶友人にアナログレコードで音楽を聴かせたら，音の膨らみ，柔らかさ，音質にしびれていました。次はPlayStationのゲーマーにX68000のゲームをさせてみようかと……。

横山　宏一(39)北海道

```
    [do] u106 L2 r>q7e a<c4>a4 b^8f8q5b8<c8 @q8d+1
    r>q7f b<d+ fe >b4<c4d4e4
    u-21q7fd >af e1 e1 d1 d1 c1 c1 r1 ff @q2e1 fq7e
    u-16q7a<c >br8u+5q3f8g8a8 u+5q7bu+5(b<e16)&e4..
    >u-10a4e4<e4d8c8
    L4 >q7f!<f!edc!>b-af! @q2g!g!8g!8q7b2 gu+5b<u+5eu+5g
/==================== Synth 2
(t6)        @5 @e80 u80 L16 o5 r4
    @p0@q1a@p+8b<@p+8c@p+8d@p+8e@p+8f@p+8g@p+8a
    @p+8b<@p+8c@p+8d@p+8e@p+8f@p+8g@p+8a@p+8b
    <@p-8c>@p-8b@p-8a@p-8g@p-8f@p-8e@p-8d@p-8c
    >@p-8b@p-8a@p-8g@p-8f@p-8e@p-8d@p-8c>@p-8b
    @p+8a@p+8b<@p+8c!@p+8d@p+8e@p+8f!@p+8g!@p+8a
    @p+8b<@p+8c!@p+8d@p+8e@p+8f!@p+8g!@p+8a@p+8b
    >>@p-8b<@p-8c@p-8d@p-8e@p-8f@p-8g!@p-8a@p-8b
    <@p-8c@p-8d@p-8e@p-8f@p-8g!@p-8a@p-8b<@p-8c
    [do] z60,+1,+1,+1,+1,+1,+1,+1,+1,+1,+1,+1,+1,+1,+1,+1
    @11 @e40 @p18 L8 o2 q4ea<cea<cea<cea<cea<ce
    o2 d+fb<d+fb<d+fb<d+fb<d+fb<d+
    o2 d+gb<d+gb<d+gb<d+gb<d+gb<d+
    o2 eg!b<eg!b<eg!b<eg!b<eg!b<e
    @9 @p72 z70,+3,+3,+3
    L16 o6 |:8q2fa<df:| |:4>ea<ce:| |:4>ea<c!e:|
    u80 L8 >q7a4.d^fa<d f!4d4c4>b4 <d2.e4 c1
    @5 @e80 r1^1^1
    u80 L24 o4 @p3@q1e@p+5f@p+5g@p+6a@p+5b<@p+5c
    @p+6d@p+5e@p+5f@p+6g@p+5a@p+5b
    <@p+6c@p+5d@p+5e@p+6f@p+5g@p+5a
    @p+6b<@p+5c@p+5d@p+6e@p+5f@p+5g
    x$40,$16,$1c, 0 @3 @e127
    u88 L16 o4 q6ga<cega<ce>cega<cega
    >>b<d+fad+fa<d+>fa<d+fa<d+fa
    >>d+egb<d+egb>gb<d+egb<d+e @p64 >a4e4<e4d8c8
    z70,+8,+8,+8,+8,+8,+8,+8
    r1^1^1 ^2>b<cdefgab
/==================== Electric piano & Pad
(t7)        i3 @128 @e127 @p64
    u100 o1 c4 ^8 i0 @82 @e127,64 @p127 r2..^1^1
    u40 o2 q8e4 @x$b6, 65,64, 5,50
    <<<u112y11,50e*1&@p0@a-50@s,32e*143@a y65,0
    [do] L1 r^^^ ^^^^
    i127 @51 @e60,0 @p74 u86 o3 @q2d^ c c!
    i0 @3 @p37 u104 L8 o5 q7a4.d^fa<d c4>b4a4g4 b2.<c4 >a1
    i127 @51 @e60,0 @p74 u86 o3 @q2d2..c >b2..a g!1 e1
    z110,-12,+4
    L8 |:q7'a<c'q6'a<c4'q7'a<c':| |:'ab'q6'ab4'q7'ab':|
    |:'gb'q6'gb4'q7'gb':| |:'fa'q6'fa4'q7'fa':|
    u102 L4 <f!<f!edc!>b-af! @q2g!g!8g!8q7b2
    eu+5gu+5b<u+5e
/==================== Flute & Oboe & Fagott
(t8)        r4 o2 u88 L4 q7a2.{aaa} b2.{bbb}
    <c!2c!{c!c!c!} u+4d2u+4g!2
    [do] @74 u74 L2 o5 |:32q8{cd}16:| u-15@q1d+u+20b
    <u+3fu+3b <u+3f1 u-9@q6d+1 u-9@q1d>b >q7{g!ab<u+4c}1
    u-5@q1df af L8 e2.de u+5b4.q7a^2
    @69 r1^1^1 ^2>u106@q1g4a4
    q7f2@q1g4a4 <d4.c>b<c>ba
    L16 <@q3e1 ^2^u-8q8dcd>u-2b<u-2c>u-2au-2b
    u100 L8 >q7a2q6ab<c>a q4fbfbr2 L4
    <q7eu-10>b<u+20bu-9a8u-9g8 u+9au-10e<u+20eu-9d8u-9c8
    @71 u100 L4 o2 q7f!<f!edc!>b-af! @q3g!1 e1
/==================== Orchestra hit & Bell
(t9)        i127 @123 @e127,20 @p64
    z105,-18 L1 o3 r4 q7a*0&<a r^^
    [do] z120,-18 >a*0&<a r >a*0&<a r >g*0&<g r >e*0&<e r
    >u105d*0&<u-18d r^^
    @51 @e60,0 @p74 u86 o2 @q2b e4.<e8d2 c2>e+2 f
    i0 @15 @e100 @p5 o4 u115 r f2a2 g! a2g2
    i127 @123 @e127,20 @p64
    z110,-18 o3 a*0&<a >a*0&<a >g*0&<g >f*0&<f
    z100,-18 >f!*0&<f! >f!*0&<f! >g!*0&<g!
    z100,-18,105,-18,110,-18,115,-18
    L4 >e*0&<e>g*0&<g>b*0&<be*0&<e
/==================== Timpani
(t10)       L32 o2 |:3u96|:7eu-2:|eue2|r4:|e4u90|:40e:|
    [do] L8 u120ar2.. ^2..{u-10aa} u|:ar2..:|
    ugr2.. ^2..{u-10gg} u|:er2..:|
    u124 <dr2.. dr2.. cr2.. r2u50|:16c!32u+2:|
    r1^1^1^1^1^1^1^1
    u124 >ar2.. ar2.. gr2.. fr4.^4.u-15f
    uf!r2.{u-15f!f!} uf!r2.. g!r2.{u-15g!g!}
    uer4.u95|:16e32:|
/==================== Bass drum
(t11)       L4 o2 u60|:8c32u+5:| |:8u110cu+15c:|
    [do] |:40u110cu+15c:|
    |:4u125c.|u-15c8ucr:|r8^2 |:8u110cu+15c:|
/==================== Snare drum & Tambourine
(t12)       L32 o2 u74|:8du-3:| |:4u50|:4du-2:||:20d:||:8du+4:|:|
    [do] L16 |:4|:6<u55f+*0>u90d8<u45f+*0>u70dd:|
    <u55f+*0>u85du-20d<u45f+*0>u65dd
    <u55f+*0>{u80du-20d}u+15d<u45f+*0>u65du+15d:|
    |:31<u55f+*0>u90d8<u45f+*0>u70dd:|
    <u55f+*0>u90du-20d<u45f+*0>u70du-10d
    |:15<u65f+*0>u90d8|<u55f+*0>u70dd:|r4.
    @x$b9, $62,38,$63,$18,6,66,$63,$1d,6,127
    |:3r2^8.^24u75e*0u100d48u85e*0u110d4:| r4u85e*0u110d4r2
    @x$b9, $62,38,$63,$18,6,65,$63,$1d,6,80
    r1 u5|:32d32u+3:| u15|:16d32u+1:||:16d32u+2:|
    <u55f+*0>u90d8<u45f+*0>u75d8<u55f+*0>u90d8<u45f+*0>u70dd
    <u55f+*0>u90du-10d<u45f+*0>u70d8<u55f+*0>u90d8<u45f+*0>u70dd
/==================== Cymbal
(t13)       L1 o3 r4 u97c+ a c+2a2 r2u10|:24a48u+2:|u+10a*0
    [do] L1 @y$18,49,48u110c+32y6,63e2.... r^ ^2
    L48u10|:23au+2:|a u100c+1 r1^1 ^2u10|:23c+u+2:|c+
    L1 u100a r u76a r u90c+ r^^^^^ ^2.
    u30|:11c+48u+3:|c+48
    L8 u95a4|:11>u56f+u40f+:| u62a+4<u95a2^u90g
    L1 u100c+ u85a u93c+ {u105c+u95au100c+g}
/==================== Electric tomtom
(t14)       L1 r4 ^^^^
    [do] r^^^ ^^^^  ^^^^ ^^^^ ^^^ ^2^8
    u90 o3c8>u+10a8u+10f8
    r^^^2<uc8>u+10a8u+10f8r8 ^^^^ [loop]
(t1,2,3,4,5,6,7,8)                      [loop]
(t9,10,11,12,13)                        [loop]
/==================== Performance
(p)
```

リスト5　ツインビーヤッホー！用カウンタ表示

```
 1:00000389 00001500   2:00000389 00001500   3:00000389 00001500   4:00000389 00001500
 5:00000389 00001500   6:00000389 00001500   7:00000389 00001500   8:00000389 00001500
 9:00000389 00001500  10:00000389 00001500  11:00000389 00001500  12:00000389 00001500
13:00000389 00001500  14:00000389 00001500
```

リスト6　闇の血族

```
================  YAMI_MAIN.ZMS  ===================
.comment [ 自己紹介 - 闇の血族 - ]  Man. by Mikio Yamazaki
.comment (C)SYSTEM SACOM
/ for Roland SC-33/55mkII/88/88VL
/ TRACK SETUP ----------------------------------------------
(i)
(b1)
(m1,2000)(aMidi10,1)     / B.D. & S.D. & Tom
(m2,2000)(aMidi10,2)     / Hats & Cymbals
(m3,2000)(aMidi11,3)     / Elec.Tom
(m4,2000)(aMidi1,4)      / Bass
(m5,2000)(aMidi2,5)      / Piano
(m6,2000)(aMidi3,6)      / Backing
(m7,2000)(aMidi4,7)      / Melody 1
(m8,2000)(aMidi5,8)      / Melody 2 & Delay1
(m9,2000)(aMidi6,9)      / Melody Delay 2
/ SC-55 SETUP ----------------------------------------------
```

▶CD-ROMドライブがほしいが，コンセント不足に悩まされています。これ以上のタコ足も怖いし……。
加藤　安弘(20)滋賀県

```
.sc55_init $10
.sc55_v_reserve $10={2,6,1,4,2,2,0,0,0,4,1,0,0,0,0,0}
.roland_exclusive $10,$42={$40,$01,$30,3} / Reverb -> Hall1
.roland_exclusive $10,$42={$40,$1a,$15,2} / Part11 -> Drum2
/ MML DATA ---------------------------------------------
(t1)    t158
/---------- Drums
/----- Bass Drum & Snare Drum & Tom
(t1)     @i$41,$10,$42
(t1)     @1 @v115 @u112 o2 q8 @p64 L4 @k0 @e60,12 r2
/--- [A]
(t1)     [do]
(t1)     |:8 cd | c8c8d :| c8d8d8d16d16
(t1)     |:8 cd | c8c8d :| cd8
(t1)     u-20 'a<c'|:3 'a8<c' :| |:4 'f8a' :| u
/--- [B]
(t1)     |:15 cd :| c8d8d
(t1)     c8c8d |:5 cd :| r8c8d c8c8d
(t1)     cdcd r8c8d cdcd cw3d cddd8d16d16
/--- [C]
(t1)     |:8 cd | c8c8d :| c8d8d8d16d16
(t1)     |:7 cd | c8c8d :| cd8 cd8c8 r.c8d8r8
(t1)     [loop]
/----- Hi-Hat & Cymbal
(t2)     @i$41,$10,$42
(t2)     @u100 o2 q8 L8 r2
/--- [A]
(t2)     [do]
(t2)     |:3 z100,70,120,85 |:4 f+f+f+ | f+ :| @u105a+ :|
(t2)     z100,70,120,85 |:3 f+f+f+f+ :| r2 <@u100c+*0&>
(t2)     |:3 z100,70,120,85 |:4 f+f+f+ | f+ :| @u105a+ :|
(t2)     z100,70,120,85 |:4 f+f+f+ | f+ :| <@u100c+ r2..u+10a>
/--- [B]
(t2)     z100,70,120,85 |:2 f+f+f+ | f+ :| @u105a+
(t2)     z100,70,120,85 |:4 f+f+f+ | f+ :| @u105a+
(t2)     |:4 z100,70,120,85 |: f+f+f+ | f+ :| @u100a+ :|
(t2)     z100,70,120,85 f+f+f+f+ r4.<@u100c+>
(t2)     |:5 z100,70,120,85 |: f+f+f+ | f+ :| @u100a+ :|
(t2)     z100,70,120,85 |:6 f+f+f+f+ :|
/--- [C]
(t2)     |:3 z100,70,120,85 |:4 f+f+f+ | f+ :| @u100a+ :|
(t2)     z100,70,120,85 |:4 f+f+f+f+ :|
(t2)     |:3 z100,70,120,85 |:4 f+f+f+ | f+ :| | @u100a+ :|
(t2)     <@u100c+> z100,70,120,85 |: f+f+f+ | f+ :|
(t2)     <@u100c+4ru-10c+r2.>
(t2)     [loop]
/----- Elec.Drum
(t3)     @i$41.$10,$42
(t3)     @25 @v98 @u111 o2 q8 @p64 L8 @k0 @e80,45 r2
/---
(t3)     [do] w rd r1 [loop]
/---------- Bass
(t4)     @i$41,$10,$42 y126,1
(t4)     @40 @v116 @u119 o2 @q2 @p64 L8 @k0 @e4,6 r2
/--- [A]
(t4)     [do]
(t4)     |:8 c :| > |:8 e :| |:8 f :| |:8 g :|
(t4)     |:8 g+ :| |:8 a :| |:8 d :| |:8 g :| <
(t4)     |:8 c :| > |:8 e :| |:8 f :| |:8 g :|
(t4)     |:8 g+ :| |:8 a :| ddddgggg |:16 a :|
/--- [B]
(t4)     |:8 f :| |:8 g :| <cccc>bbbb |:8 a :| <
(t4)     |:8 d :| > |:8 g :| < |:16 c :| > |:8 f :|
(t4)     |:8 g :| <cccc>bbbb aaaagggg |:8 f :|
(t4)     |:8 g :| < |:16 c :| >
/--- [C]
(t4)     |:16 f :| |:8 e :| |:8 a :| |:8 f :|
(t4)     |:8 g :| < |:16 c :| |:16 c+ :| |:16 d :| >
(t4)     |:16 a+ :| |:7 g :| g& gggr2.
(t4)     [loop]
/---------- Piano
/ SC-55,CM-300/500ユーザーの方は@6にしてください。
(t5)     @i$41,$10,$42
(t5)     i16 @5 @v109 @u108 o4 @q1 @p64 L8 @k0 @e50,20 r2
/--- [A]
(t5)     [do]
(t5)     |: 'e2.g<c''eg<c''egb'& 'e2^8gb''egb'r'egb'
(t5)        'c2.fa''cfa'>'b<dg'& 'b2^8<dg''b<dg'r'b<dg'
(t5)        'b2.<eg+''b<eg'<'cea'& | 'c2^8ea''cea'r'dfa'&
(t5)        'd4fa''d4fa''dfa''dfa'r>'b<dg'&
(t5)        'b4<dg''b4<dg''b<dg''b<dg'r'b<dg' :|
(t5)     <'c4ea''c4ea''cea''cea'r'cea'
(t5)     'd4fa''d4fa''dgb''dgb'r'ea<d'&
(t5)     'e4a<d''e4a<d''e4a<d''ea<d''ea<c+'& |:8 'ea<c+' :|
/--- [B]
(t5)     L1
(t5)     |: 'cfa''dgb''e2g<c''d2gb' | 'cea'
(t5)        >'a<df''b<dg'<'cfg''ceg' :| 'c2ea'>'b2<dg'
(t5)     'a<cf''b<dg'<'cfg''ceg'
/--- [C]
(t5)     L8 >
(t5)     'a2.<cf''a<cf''b8^1<dg'
(t5)     'b2<dg''b4<dg+''b<dg+''<c8^1ea'
(t5)     'a2^8<df''a<df'r'a<df' 'b1<dg'
(t5)     <'c2.fg''cdg''ceg'& 'c2^8eg''ceg''cfg''ceg'
(t5)     'c+2.ea''c+ea''c+ea'& 'c+2.ea''c+4ea'
(t5)     'd2.fa''dfa''dfa'& 'd2^8fa'>'a<df''a<df''a+<df'&
(t5)     'a+2.<df''a+<df''a+<df' 'a+4.<df''a+4.<de''a+<df'
(t5)     <'c2..dg''cdg''cdg'& 'cdg''cdg''>b<dg' r2.
(t5)     [loop]
/---------- Sequence
(t6)     @i$41,$10,$42
(t6)     @13 @v108 @u104 o5 @q1 @p74 L8 @k0 @e70,20 r2
/--- [A]
(t6)     [do]
(t6)     |: rg<cdgdc>g rb<deged>b ra<cfgfc>a rb<dgagd>b
(t6)        rb<deg+ed>b r<ceabaec> | rfa<cdc>af rb<dgagd>b:|
(t6)     rf<cfgfd>g ra<deaed>a ra<eabae>a
/--- [B]
(t6)     |: f<cfc>f<cfc> g<dgd>g<dgd> g<cgcg>b<g>b ea<e>aea<e>a
(t6)     |  da<d>ada<d>a dg<d>gdg<d>g cg<c>gcg<c>g cg<cggcgc>:|
(t6)     cf<c>fcf<c>f dg<d>gdg<d>g cg<c>gcg<c>g <cg<ggc>>r4.
/--- [C]
(t6)     r*3096 [loop]
/---------- Melody
/ (t7)~(t9)まで、[A]と[B]はほとんど同じです。
/----- Main
(t7)     @i$41,$10,$42
(t7)     @q1 L8 @k0 @e50,60 r2
/--- [A]
(t7)     [do] @82 @v110 @u102 o5 @p74 @e,60
(t7)     e2.cd4r>bargrba2.<c>b2..r
(t7)     g<e4ferdrce4>aa2^rab<c4.ee4r dr>barg<
(t7)     e2.gd4r>bargrba2.<c>b2..r
(t7)     g<e4ferg+rabaecr>bra4r <crdr>ba^2^1
/--- [B]
(t7)     @63 @v115 @u108 o5 @p64
(t7)     derf^2>a4ab4.<c4d>brg4.f4fgfe^2
(t7)     a4>b<c2.>b<cf4fergrdd2..cg^2
(t7)     <derf^2>a4ab4.<c4d>brg4.<c4d4cg^2
(t7)     c>b4a^2<c4ag4.d4c4>b4<d2d4dc2.
/--- [C]
(t7)     @69 @v105 @u102 o4 @p64 @e,30
(t7)     b<c>ba2<c4>ab^1 b2<e4ff ree4.ede
(t7)     f4ferd4d f4ferd4c d2.fe2.efg
(t7)     g2e4a+a+2a4.efg^2fef2. fed2.
(t7)     deff4.e4.fg2..gg4ggr2.
(t7)     [loop]
/----- Layer & Sub
(t8)     @i$41,$10,$42
(t8)     @q1 L8 @e30,80 r*97
/--- [A]
(t8)     [do] @82 @v103 @u98 o4 @p54 @k2 @e,80
(t8)     e2.cd4r>bargrba2.<c>b2..r
(t8)     g<e4ferdrce4>aa2^rab<c4.ee4r dr>barg<
(t8)     e2.gd4r>bargrba2.<c>b2..r
(t8)     g<e4ferg+rabaecr>bra4r <crdr>ba^2^1
/--- [B]
(t8)     @63 @v105 @u98 o5 @p84 @k6 @e,60 r8
(t8)     derf^2>a4ab4.<c4d>brg4.f4fgfe^2
(t8)     a4>b<c2.>b<cf4fergrdd2..cg^2
(t8)     <derf^2>a4ab4.<c4d>brg4.<c4d4cg^2
(t8)     c>b4a^2<c4ag4.d4c4>b4<d2d4dc2^8
/--- [C]
(t8)     @69 @v104 @u96 o4 @p64 @k1 @e,30
(t8)     gagf2a4fg^1 g2b4<dd rcc4.c>b<c
(t8)     d4dcr>a4a <d4dcr>b4a b2.<dc2.cde
(t8)     e2c+4gg2e4.c+de2^dc+d2. dc>a+2.
(t8)     a+<cdd4.c4.dd2..dd4ddr2.
(t8)     [loop]
/----- Delay
(t9)     @i$41,$10,$42
(t9)     @q1 @p44 L8 @k-4 @e80,40 r2 r8
/--- [A]
(t9)     [do] @82 @v80 @u82 o5
(t9)     e2.cd4r>bargrba2.<c>b2..r
(t9)     g<e4ferdrce4>aa2^rab<c4.ee4r dr>barg<
(t9)     e2.gd4r>bargrba2.<c>b2..r
(t9)     g<e4ferg+rabaecr>bra4r <crdr>ba^2^1
/--- [B]
(t9)     @63 @v95 @u88 o5 @k-6 r8
(t9)     derf^2>a4ab4.<c4d>brg4.f4fgfe^2
(t9)     a4>b<c2.>b<cf4fergrdd2..cg^2
(t9)     <derf^2>a4ab4.<c4d>brg4.<c4d4cg^2
(t9)     c>b4a^2<c4ag4.d4c4>b4<d2d4dc2^8
/--- [C]
(t9)     @69 @v86 @u92 o4 @k-4
(t9)     'gb''a<c''gb''f2a''a4<c''fa''g8^1b'
(t9)     'g2b''b4<e'<'df''df' r'ce''c4.e''ce''>b<e''ce'
(t9)     'd4f''df''ce'r>'a4<d''a<d' <'d4f''df''ce'r>
(t9)     'b4<d''a<c' 'b2.<d'<'df''c2.e''ce''df''eg'
(t9)     'e2g''c+4e''ga+''g2a+''e4.a''c+e''df''e8^2g'
(t9)     'df''c+e''d2.f' 'df''ce''>a+2.<d'
(t9)     '>a+<d''ce''df''d4.f''c4.e''df''d2..g'
(t9)     'dg''d4g''dg''dg'r2.
(t9)     [loop]
(p)
```

▶ 9月号の特集を見て，8年前のマシンでこれだけの映像が映し出されるのかとX680x0のすごさを再認識しました。動いているところを見てみたいです。須永　悦啓(24)栃木県

リスト7　闇の血族用カウンタ表示

```
1:00000060 000024D8   2:00000060 000024D8   3:00000060 000000F0   4:00000060 000024D8
5:00000060 000024D8   6:00000060 000024D8   7:00000060 000024D8   8:00000061 000024D8
9:00000078 000024D8
```

リスト8　TIME STREAM

日本音楽著作権協会（出）許諾第9571367-501号

```
===============  TIME_STRM_投稿版.ZMS  ==================
.COMMENT -- TIME STREAM -- CASIOPEA  for SC-88  by  ﾏｯﾁｭﾝ
(I)(B1)
(M1,4800)(AMIDI1,1)
(M2,3700)(AMIDI2,2)
(M3,6700)(AMIDI3,3)
(M4,6810)(AMIDI3,4)
(M5,1800)(AMIDI5,5)
(M6,6400)(AMIDI6,6)
(M7,6400)(AMIDI7,7)
(M8,4800)(AMIDI8,8)
(M9,4800)(AMIDI9,9)
(M10,5830)(AMIDI10,10)
(M11,3000)(AMIDI11,11)

.ROLAND_EXCLUSIVE $10,$42={$00,$00,$7F,$00}  /音源初期化
.ROLAND_EXCLUSIVE $10,$42={$40,$11,$34,$3F}  /attack
.ROLAND_EXCLUSIVE $10,$42={$40,$12,$34,$3E}  /attack
.ROLAND_EXCLUSIVE $10,$42={$40,$13,$34,$3D}  /attack
.ROLAND_EXCLUSIVE $10,$42={$40,$15,$34,$3D}  /attack
.ROLAND_EXCLUSIVE $10,$42={$40,$18,$34,$42}  /attack
.ROLAND_EXCLUSIVE $10,$42={$40,$19,$34,$3E}  /attack
.ROLAND_EXCLUSIVE $10,$42={$40,$11,$32,$44}  /cutoff freq
.ROLAND_EXCLUSIVE $10,$42={$40,$12,$32,$48}  /cutoff freq
.ROLAND_EXCLUSIVE $10,$42={$40,$19,$32,$45}  /cutoff freq
.ROLAND_EXCLUSIVE $10,$42={$40,$11,$33,$3D}  /resonance
.ROLAND_EXCLUSIVE $10,$42={$40,$19,$33,$3A}  /resonance
.ROLAND_EXCLUSIVE $10,$42={$40,$11,$36,$42}  /release
.ROLAND_EXCLUSIVE $10,$42={$40,$18,$36,$42}  /release
.ROLAND_EXCLUSIVE $10,$42={$40,$19,$36,$42}  /release
.ROLAND_EXCLUSIVE $10,$42={$40,$1A,$15,$02}  /Use Drumpart


.sc55_print "TIME STREAM"

.SC55_reverb $10={4,4,0,72,122,0,0}             /Reverb set

.SC55_chorus $10={2,0,68,12,78,04,20,00}        /Chorus set


.ROLAND_EXCLUSIVE $10,$42={$40,$01,$50
          4,0,$61,4,4,127,0,0,64,80,0}             /DelaySet

.ROLAND_EXCLUSIVE $10,$42={$40,$02,$00,
               $01,$3D,$01,$45}                   /EQ set


(t1)t72o6@v100q8@p72@g12@u120l16i0,1@30k-12@j1    / Guitar
            m2,1h0,0@h12,0s2,3@s3,19@c11,127,127

(t2)o4q7@v78@p45@g12@u80l32i2,2@100                / Harp.
            m,1h,1@h,0s,2@s,42@c10,64,127

(t3)o4q7@v127@p52@g12@u80l16i16,2@1                / Pf
(t4)o3q7@g12@u70l16                                / Pf low

(t5)o5q7@v114@p46@g12@u88l16i8,2@55                / ﾕﾆｿﾞﾝ
            m2,1h0,0@h12,0s2,3@s3,32@c11,127,127

(t6)o1q7@v114@p64@g12@u110l16i3,2@36@j1            / Bass
            m2,1h0,0@h0,192s2,3@s5,84@c11,127,127
(t7)o1q7@v105@p64@g12@u110l16i5,2@36@j1            / Bass
            m2,1h0,0@h1,192s2,3@s4,84@c11,127,127

(t8)o5q7@v100@p86@g12@u88l16i8,2@55                / Voice
            m2,1h0,0@h12,0s2,3@s3,32@c11,127,127

(t9)o6@v110Q8@p56@g12@U120l16i0,2@30k-12@j1        / Guitar
            m2,1h0,0@h12,0s2,3@s3,19@c11,127,127

(t10)o3@v105@r1@g12@u100l16i0,2@41                 / Drum
(t11)o2@v100@r1@g12@u127l16i0,2@41                 / Drum


(t1)@I$41,$10,$42@E62,32  y94,19
(t2)@I$41,$10,$42@E40,15  y94,0w
(t3)@I$41,$10,$42@E65,18  y94,6
(t5)@I$41,$10,$42@E40,55  y94,4w
(t6)@I$41,$10,$42@E10,22  y94,0
(t7)@I$41,$10,$42@E12,32  y94,0
(t8)@I$41,$10,$42@E75,45  y94,0w
(t9)@I$41,$10,$42@E59,40  y94,25
(t10)@I$41,$10,$42@E65,32 y94,0 @y$18,38,$41   / Drum ﾋﾟｯﾁ
(t11)@I$41,$10,$42@E22,48 y94,0

/------------------  ｲﾝﾄﾛ  ------------------------------

(t1,9)
r1@a-105o6  r2r@u90d*4&e*32&@b0,-3415,36q6e4@b0
@a-88q6@u104c*4&@m62d*188@m
>q7@u92b*4&<@m84c*92@m>z110,,104,104a*4&b*44g*4&a-*44
@u99f*4&g*92@u84@m121q5e2@m@a

(t3)
r1  o4q7@u94r*9'ef+<c'*135,3z110,95,64{bae}4
z94,88,76r*9g*135f8e8  z86,76,88,95,105f4.e8d8.e8.f8>
z95,120,105,90,88'g4<de',0ra<dg>r*3'b<eg+'*93,3

(t4)
r1o2q7  @u105'a1<e<c',3  z98,87@d1'a2.<f<c'<a4>@d0
z100,88,94'g2<f<ce',0e*3'e<fa-b'*93  r4z110,100<g4f2>

(t6,7)
r1  r1  @u100o1r2.@b0,4781,12@q2d4@b0  @u110d1
z100,95,90,100c4.@b0,2049,12g8@b0@b0,-4098,24q6b-4@b0q7e4

(t10)
r1o3  z95,100,90,105,95,80d+4d+4d+4{d+d+d+}4
z92d+4z50,55,60,65,70,75,80,85{d+d+d+d+d+d+d+d+}4
z90,85,80,75,70,65,60,55,50,45,40,35,30,25,20,15
{d+d+d+d+d+d+d+d+d+d+ d+d+d+d+d+d+}2
z95,100,95,100d+4d+4d+4d+4
z95,105,95,80,46,70d+4{d+d+d+}4a4d+4

(t11)
r*960

/---------------------  A  ----------------------------

(t1,9)
o5q8z115,102,122,,98,108,,98e8a8b*1&<c*23>aa*2&b*82a8
z95,100,88,108,,95q8@b0,-1366,36g4@b0q8f8ee&@a50@m64f4.
@m@a>g8<@u115f*4&g*44&f8z88,108,100,,88ef16^4@b-3415e*4&
@b0e*20d@u101e&q8@m64@a54e2@a@mr2q8z115,102,122,,98

e8a8b*1&<c*23>a@u110a*2&b*10&@b0,-1366,60b4.@b0@u98a8
z105,100,88,108,,112g4f8ee&@a50f4.@ag8
@u119a-*3&b-*45&a-8&g@u110a-4^16@u119b-*2&b*46
@u110b-8&g@u100b-&@a52@m88q5b-2@a@mq8z115,102e-8f8


(t3)
o4q7z88,98'c8ef+',0'cf+8a'z88,60'c8ef+'>a8<'c8ef+'>a8<
'c8ef+'>a8<'c8dg'>a8<'c8dg'>a8<'c8dg'>a8<'c8dg'>a8<
'c8df'>a8<'c8df'>a8'b8<df'a-8'b8<df'a-8
z92,88,82,81,82,78,89<'c8d'>q8ge8g<e>b<'e2g+b',3

o4q7z88,98'c8ef+',0'cf+8a'z88,60'c8ef+'>a8<'c8ef+'>a8<
'c8ef+'>a8<'c8dg'>a8<'c8dg'>a8<'c8dg'>a8<'c8dg'>a8<
'c8e-g'>a-8<'c8e-g'>a-8<'c8df'>a-8<'c8df'>a-8<
'c8df'>g8rz82,92,102,95q8g<cf>'a-<d-'a-<d-f'd-4e-'


(t4)
@u100q7o2'a1<e'  'a1<f'  'g1<f'  r1  'a1<e'  'a1<f'
z105,110,92,100,88r2b-2  r8.q6<e-q7e-4>'b-4<f''a4<g'


(t6,7)
o1z108,92q7a2...q5a  @u108q6a1  g1 z100,112,102<c2q8>b-4e4
z108,92q7a2...q5a  z108,80q7a2^8.q2aq7a8.q2a
z110,80,100q7b-2^8.q2b-q6<@b0,-8191,24f4>@b0
z100,102,92q6e-2q6b-4a4q7


(t10)
@u85o2|:3d4d4d4d4:|  d4d4@u46<a2>  @u85|:4d4d4d4d4:|

(t11)
z100,80,80o2|:3c4g+2g+4:|  c4g+2.
z100,80,80o2|:3c4g+2g+4:|  z100,80,88c4g+2a+4

/---------------------  B  ----------------------------

(t1,9)
o5z112,,104,116,,89,102,,109q8@b-3415g*3&@b0g*21<c8
d*1&e-*23q7cq8c&@m33@a20d4.@m@aq7e-8z92,,88,78,,125,,102>
q8b-*2&<c*22>q7b-q8b-&@m88@a55b-2@m@a@b3415e-*3&@b0e-*21f8
z112,104,110,,89,102,,119,119q8g8<c8d*1&e-*23q7cq8c&
@m33@a20d4@m@arq6@b-3415f*2&@b0,-8191,20f*34@b0

o5q8r*14z121,,94,82,125,,102b-*2&<c*18>q8b-*14q8@a45@m84
b-2@m@a@b3415e-*3&@b0e-*21f8z112,104,122,,89,102,,109
q8g8<c8d*1&e-*23q7cq8c&@m33@a20d4.@m@aq7e-8z100,112,95o5
q6@b0,-3415,24b-4@b0q8g8b-8@u95<c*2&@m0,128,,,,,,128d*70&

@mc8>z95,90,96,86,,100,,98b-4a-8q7gq8a-&@m69@a33a-4@m@a
f*3&g*21q7fq8@u88g&@m95@a38g2@m@az116,102r4g8b8


(t3)
z95,112,122,115,125q7o4'c4e-''c4e-g''d4fb-'r>b-<d8>
z100,88,102,110,115'b-4<e-'r<e-d8d-4d-4z125,115,105,95,110
c8g8<c8d8>'f4a-<c'q8>b-<dfa-q7z110,110,120,92,110,118>

'b-4<e-'rq8b-<e-dq7d-4d-4z116,95,88,118,112,102,115,102
o5c8.>cb-*2<'cg'*22q8c>gq7'd4^16f'>b-8.z110,105,95,78,112
'b-4.<e-'<q8{rfdc}8>q7'b-2<d'z110,120,105,90,109,86

o3'g4a-<c'rq8ga-<e->q7'a-4..<c'<e-z114,108,98,88,100,110
'c4df'rq8>a<cdq7'c4dfa'r*3'dfgb'*45


(t4)
@u100o2q7'a-2<g'r2  'g2<f'z108,102,95,105,95'b-4<a-''a4<g'
'a-2<g'<<d8.>a-r4z100,108,102o2'g2<f''b-4<a-''a4<g'
```


▶ 8月21日(月)，晴れ。運転免許証の更新に行きました。1：更新手数料が2,900円。2：交通安全協会費が5年で2,500円。2の存在を忘れていたので，お金がギリギリでした。

春名　義行(28)兵庫県

```
o4r4z110,102,100,108g4c4r4  o2'g2<f'<'c2f'>
z102,112,108,100,92,100'f2<f'<e-2 d2>'g4<g''f<f'*3'f<f'*45

(t6,7)
z110,95,105q7a-4r8q5a-8q7a-2
z110,95,108,100g4r8.q5gq7b-4a4
z110,95,100,80,80a-4r8q5a-8q7a-4r8q2a-a-q7

z110,95,115,105g4.q5(b-8<c),12q7>b-4a4
z110,95,100,80,80a-4r8q5a-8q7a-4r8q2a-a-
z110,95,105,110,,100,90q6g4q5{r<dc+}4q7c4&{c>gf+}4

q7z110,105,95f2e-4..q5e-q7
z110,105,95,110,100d4f8q5f+8q7g4f4

(t8)
r*1344  o4q8z100,95,88f2'c4fa''d4gb'

(t10)
@u45o3a4>@u85d4d4d4  |:4d4d4d4d4:|  d4d4@u75<a2>
@u85d4d4d4d4  d4d4<z95,100d+4d+4

(t11)
w8z100,80|:5c4g+4c4g+4:|  c4g+4c2
z100,80,,100,88,,88c4g+2g+4  c4a+4a+4a+4

/----------------------  C  ------------------------------

(t1,9)
q8o6@u120c*4&@m0,88,,,,,,88@a35d*116@m@aq8z88,105,95
c8>e8q7g8z112,,108,100,100@b-3415b*2&@b0b*22q8aq7b&
@m64@a38b2@m@az126,,102@b-3415a*3&@b0a*21<c8
q8o6@u118d*4&@m0,88,,,,,,88@a35e*116@m@aq7z115,82,108,108
d8.q8cq7d*3&e*21z122,102,95,,120,105q8d8q6cq7d&
@m0,121,121,64,0,0,0,0@b0,-2049,68@a30d2@m@a@b0q8>g8b8

q8o6@u120c*4&@m0,88,,,,,,88@a35d*116@m@aq8z115,95,100
c8.>q6e8q8gz115,,100,88,88q7@b-3415b*3&@b0b*21q8aq7b&
@m64@a15b2@m@az125,,105q8@b-3415a*3&@b0a*21<c8q8o6@u120
d*4&@m0,88,,,,,,88@a35e*116@m@aq7z105,92,100q8r*4d8c*20e8
@u112q6f*3&@m0,124,212,124,64@a54g*141@a@mr4

(t2)
@a127o5z110,100egz90,80<c>b<c>b<c>b<c>b<c>b<c>b<c>b<c4@ar4
@a127z75o5'egb<d'z80,90<cdcdcdcdcdcr8@ar2@a127
z110o4az100,90,80,70b-<dfab-<dfa>b-<dfa>b-<dfa4@ar4@a127o5
z110,100fa-z90,80<c>a-<c>a-<c>a-<c>a-<c>a-<c>a-<c>a-@ar2

@a127<c>b<c>b<c>b<c>b<c>b<c>b<c>b<c>b@ar2@u78o6@a127
eeeeeeee eeeeeeee@ar2 @a127z90,80o5afafafaf afafafaf@ar2
o4z110,105,100,95,90,85,80,75
@a127fb-<ce-fb-<ce- e-c>b-fe-c>b-f ga-<ce-ga-<ce-
z70,65,60,55,50,45,40,35eeeeeeee@aw

(t3)
z120,108,98,100,95,88o4q7r4'g8<g'e>b<dg'e8<e'r4
z88,110,117,112'e4gb<d'r*18q8'egb<c'*16'egb<d'*14q7'e2gb'
r4..<z115,127q5'da<d'q7'd2a<d'z88,99,110,115
o3g+*3<'a-<cdf'*93r*4'cfa-<c'*44'd4gb'

o5r8z100,108,119,105,,98q8r*10'cgb'*14'cgb'*18'da-<c'*14
'cdgb'*16&q7'c4dgb'r8.>'d-fa-<c'
z110,78,120,110,100,90,110,120'c4egb'rq8a*2>'b<b'*10
<ceg<d>q7b4.z100,127,88'e<e'*18'e<e'*14a*16'e<e'*18
'e<e'*14a*16'e4<e'z110re8.
z110,100,80,105o4q7'g4..b-<df'<<q8c*6>a-*6q7'c2dg'

(t4)
o3q7@u105'b1<de'  r1  @u110'b-1<fa'
r8z95,110,95,85,110,120g+g+4^16q8f*2f+*2q7g+*44f+4
@u110'g1b<d'  r*576

(t6,7)
o1q7z110,116,96,120,80,100e4^16b8q5<ebq7eb4.>
z110,95,105,80,80a4..q5aq7a4.q2aaq7
z110,100,92,115,100,88,98b-4^16<q5(e8f),12q7b-<e>b-<q8e8.
@b0,-8191,12q6e8.q7@b0o1q7z110,95,110,115a-4..q5a-q7g4f4

o1q7z110,116,96,120,80,100e4^16b8q5<ebq7eb4.>
z110,95,100,115,125,95a4..q5aq7a4rq8<eaq5eq7>
z110,100,92,115,100,110b-4^16<q5(e8f),12q7b-<e>b-
<@b0,-8191,48q6e4.q7@b0  @u110q7o1e-2e2

(t8)
@u78o4q8'd1egb'  'e1gb'  >'b-1<de'  <'c2.fa-<c''d4fgb'
'd1egb'  'c1egb'  >'b-1<dfa'z95,88<'f2b-'@a80'g2<c'  r1w

(t10)
o3z100,95,108,100,94|:7d+4{d+d+r}4d+4d+4:|
d+4{d+d+r}4@u58a2

(t11)
w2z100,80c4g+4c4g+4  c4{g+rc}4z100,80c4g+4  c4g+4c4g+4
c4g+4c4{g+rc}4  z100,80c4g+4c4g+4  c4{g+rc}4z100,80c4g+4
c4g+4c4g+4  c4{g+rc}4@u100c2

/----------------------  D  ------------------------------

(t1,9)
o5q8z115,102,122,,100,108,,88e8a8b*1&<c*23>aa*2&b*82a8
z95,100,88,108,,88q6g4q8f8ee*3&@a50@m64f*81@m@a>
@b0,3415,12@a32q6g8@a@b0<z114,100,88,108,,120,,105,95,95
g4q8f8e8q8f8&a-8q7<d*2&e*22q8dq7d&  @a66@m68e2.@a@mr4

(t3)
o4q7z88,98'c8ef+',0'cf+8a'z88,60'c8ef+'>a8<'c8ef+'>a8<
'c8ef+'>a8<  'c8dg'>a8<'c8dg'>a8<'c8dg'>a8<'c8dg'>a8<
'c8df'>a8<'c8df'>a8'b8<d'a-8<f8>a-8
z88,110,120,127,88,80,110,113,116,119,122,125,127
q7'b8.<e''g8<g''b8<a'q8<b*8g+*4b>q5b<ef+g+ab<c

(t4)
o2q7@u110'a1<e'  'a1<f'  z110,120,115'g2<f'<f2  c1

(t6,7)
o1z108,80q7a2^8.q2aq7a8.q2a
z110,92,92q7a2^8.(g8a),12&q6a8.
@u110q7g1  z110,92,100,115<q7c4..q5cq6>b-4q7e4

(t10)
o2@u85|:3d4d4d4d4:|  d4d4@u48<a2>

(t11)
z100,80,80o2|:3c4g+2g+4:|  z100,88,90,100c4{a+rc}4c2

/----------------------  E  ------------------------------

(t1,9)
@a-95q8o6@u114@m64d2.@mr4  @u102c2&>@m64a2@m
z121,114,102,102g4e4f4&@m64a-4@m  @u108g2&@m88e2@m
@u100d4&>@m64b2@m@u95<c4&  >@m69a2.@mr4
@u106@m64b-2@m@u115<f4&@m62d4@m
@u100>@m65b-2@m@ar4<z125,102e-8f8

(t3)
q8o5z85,108,73,65,80,57,45>b*2<c*10>b<c*2d*10c>q7a4
z80,100,110,115{rab<cc}4@u127q4{cccccz117,110cc}4

z80,110,95,100,114,92,78>b*2<c*10>b<c*2d*10c>q7a4r8.
z115,70,80,90,100,110,120gq8{a<cdefg}4
z127,90,123,88,118,84,112,80g*17f*7f*16e*8e*15d*9d*14c*10
z100,110,120,115,110,105>b*10<c+*9d*8c+*7>b*7a*7
z100,95,90,85,80,90,100a-*9g*8f+*7f*6d+*6e*6g*6
z80,95,85,119,90,80,70q8c*10r*6g*8a*6<d*6>b*6a*6g4
z100,70,120,100,80,127,107,87,67,110,90,70
o3g+<{g+<e>b}16<{g+<e>b}16{g+eb}16g+*6e*9>
z70,75,80,85,90,95b*4<c*5c+*6d*6d+*6e*6

z100,105,110,115,120,127,100,80,95,70
r*10f+*7g+*7a*8b*8<c*8 dc>b<c>a4rz127,110aq6b8q7
q6z105,127,80'cfg<c'*18'cfg<c'*14'cfgb'*16'cfa'*18'cfa'*14
'cg'*16>z90,100,65'f<cf'*18'e<ce'*14'f<cf'*16
z120,100,110,80q7'g<g''g<g''a-<a-''b-<b-'
q7z110,116,124,95,127o5r*18'c<c'*14'd<d'*16'e-<e-'*18
'f<f'*12'f+<f+'*18q8
z110,127,100,120,90,110,80,100,70,90,60,80,50,70,40,50
{g<g>g<g>g<g>g<g>g<g>g<g>g<g>g<g>}4
z125,80o5q7'f<f'*18'e-<e-'*30

z110,125,95,100,120,100,114'd<d'*18'f<f'*14'c<c'*16q6
'd4<d'q7'd-8<d-''e-8<e-'r*4'e-<e-'*44

(t4)
r1  o4q7r4..@u95'e2^16fa'  z118,112>'g2<cf''f2b<e'
@u120r*10'e<d'*38r2.  q7r4z120,100,80,60,110,82,120,80
'a<ce''a<ce''a<ce''a<ce''a8.<ce'<ce8.>az100,127,80
'a<cg'*18'a<cg'*14'a<cg'*16'a<cf'*18'a<cf'*14'a<cg'*16
z90,100,72'a<c'*18'a<c'*14'a<c'*16z110,100,90,80
'a<c''a<c''a<c''a<c'z100,127,80'a-<cg'*18'a-<cg'*14
'a-<cg'*16'a-<cg'*18'a-<cg'*14'a-<cg'*16
z80,100,70,90,60,80,50,70,40,60,30,50,20,40,10,30q8y66,127
{'a<b-<d'<f>'a<b-<d'<f>'a<b-<d'<f>'a<b-<d'<f>'a<b-<d'<f>'a<b-<d'<f>'a<b-<d'<f>'a<b-<d'<f>}4

y66,0r4q7z110,127,100,124,104,84,64,117,100,80,114o3
'gb-<df'*18'gb-<df'*14'gb-<df'*16q8'a<f'<f>b-q6a-
q7'g4b-<d-f'q8g*2<d-*2q7e-*44

(t6,7)
o1z108,80q7a2^8.q2aq7a8.q2a
z110,95,110,110,115,105,80,90,100q7a2...q5aq7  g1
<c4d4>b-4<q2eq5>b-q6e8  z110,95,115,120
q7a2^8.q5bq7<f+8.q5b  z110,95,100,80,110,80
q7a4..q5eq6(d8c),12rq2c>q7a8.q2a
z108,80q7b-2^8.q2b-q7b-8.q2b-
z110,115,110,100<q7e-4.q5f8q6>b-4q8a4

(t10)
o2@u85|:3d4d4d4d4:|  d4d4@u48<a2>  @u85|:4d4d4d4d4:|

(t11)
z100,80,80o2|:3c4g+2g+4:|  z100,80,100,88c4g+4c4a+4
z100,80,80o2|:3c4g+2g+4:|  z100,88,88c4a+2a+4

/-----------------------------------------------------------
/----------------------  repeat  ---------------------------
/-----------------------------------------------------------

/-------------  B  - toCODA1 + CODA1  ----------------------
```

▶040turboを載せたのに，やっていることは音楽の打ち込みとTeXばかり。私のコンピュータが真の力を発揮するのはいつだろうか。　福間　雅弘(17)大阪府

```
(t1,9)
o5z112,104,116,,89,102,,109q8g8<c8d*1&e-*23q7cq8c&@m33@a20
d4.@m@aq7e-8z92,,88,78,,125,,102>q8b-*2&<c*22>q7b-q8b-&
@m88@a55b-2@m@a@b3415e-*3&@b0e-*21f8
z112,104,110,,100,92,92q8g8<c8d*1&e-*23q7cq8c&@m33@a20
d4@m@ar*14@u119e-*3&f*11&c*2@u88f*18

o5q8z111,,104,98,125,92r*10b-*2&<c*12>b-@a45
@m0,84,,,,,0,0q7@b0,-3415,48b-4^16@m@a@b0r4e-8f8
z112,104,122,,89,102,,109q8g8<c8d*1&e-*23q7cq8c&
@m33@a20d4.@m@aq7e-8z100,112,,95o5q6@m64b-4@mq8
@b-3415g*3&@b0g*21b-8@u108<c*2&@m64d*70&@mc8>

z95,90,96,86,,100,,98b-4a-8q7gq8a-&@m69@a33a-4@m@a
f*3&g*21q7fq8@u88g&@m95@a38g2@m@az75,85,95,105,115
r8.q8ga-<ce-f@u116q8f*3&@b0,-8191,80@m0,69,,,,,0,0@a51g*93
@m@b0@ar4q8>z125,102g8b8

(t3)
q7o4z108,105,95,75,85,70,90,105,115'a-8.<ce-g',3q8'ce-g',0
>a-a-<c>a-q7'b-4<df'b-<f8.z100,108,90,98,85
>'b-4<e-'rq8b-q7b-8<'d-4f''d4e-'z100,95,112,85
'c2e-'r8.q6'a-b-<dfb-''a-b-<dfb-'q7'a-8.b-<dfb-'

o3z90,78,100,110,90,70,85,105,90q8f*2b-*2q7'b-<e-'*44
q8rb-<e-dq7d-8q8fq7d-4^16>z100,110,90,70,50,95,78,90
'g4<c'<<q8'cg'e-c>gb-q7f8.'f4<f'z100,110,90,70,90
e-4.q8df*6d*6q7>b-2'a-4<ce-'rq8a-<c>a-<q7e-2>

z100,117,70,100,111,95,85'a4<cd'<r*18'a<ce'*14'a<d'*16
'a8<ce'<q8fq7>'a8.<ce''a8<c'z100,110,90,70,78,80,88
'f8<c'q8rcfaq7<c8>'d4gb'r*3'fb<d'*45,2

(t4)
o3q7z95,110,100ra-a-8r4a-2>z100,110,95,100
'g2<f''b-4<a-''a4<g'  'a-2<g'r2  z100,110,95
'g2<f''b-4<a-''a4<g'  z100,90,114,110,100
'a-8.<e-'<<e-g4>'a-2b-<d'  g4r2.  z110f2r2
o2z110,124,80,110,105,95f4<r*18'f<c'*14'f<c'*16'f8.<c'
'f8.<c''f8<c'z95,100,110,100,115q8f*3g*21q7
y64,127a4.y64,0>'f4<f'<g4

(t6,7)
o1z110,95,105q7a-4r8q5a-8q7a-2
z110,95,108,100g4r8.q5gq7b-4a4  z110,95,100,80,80
a-4r8q5a-8q7a-4r8q2a-a-q7  z110,95,115,105

g4.q5(b-8<c),12q7>b-4a4  z110,95,100,80,80
a-4r8q5a-8q7a-4r8q2a-a-  z110,80,80,115
q6g4r8q2ggq8@b0,8191,48c2@b0  q7z110,105,95f2e-4..q5e-q7
z110,95,120,110,100d4.q5f8q7f8.g8.a8
z110,100,120,95,114,120g8.<@b-3415,0,1d*4&@b0d*20q6gq7d8
q6f4@b0,8191,24c4@b0q7

(t8)
r*1344  @a-96o5q7@u110r2c2&  c2>@u88@a'd2gb'

(t10)
@u45o3a4>@u85d4d4d4  |:2d4d4d4d4:|  @u45o3a4>@u85d4d4d4
d4d4d4d4  d4d4@u75<a2>  @u85d4d4d4d4  @u45o3|:2a2c+2:|

(t11)
w8z100,80|:5c4g+4c4g+4:|  z100,88,100c4a+4c2
z100,80,88c4g+2a+4  z100,88|:2c4a+4c4a+4:|

/--------------  C  - toCODA2 + CODA2  -------------------

(t1,9)
q8o6@u114c*4&@m0,88,,,,,,88@a35d*116@m@aq8z88,105,95
c8>e8q7g8z112,,108,100,100@b-3415b*2&@b0b*22q8aq7b&
@m64@a38b2@m@az126,,102@b-3415a*3&@b0a*21<c8
q8o6@u118d*4&@m0,88,,,,,,88@a35e*116@m@aq7
z115,82,108,108d8.q8cq7d*3&e*21z122,102,95,95q8d8q6cq7d&
@m0,121,121,64,0,0,0,0@b0,-2049,36@a25d4@m@a@b0r8.q8>
z115,125,115,105dq7{egb}4

q8o6@u114c*4&@m0,88,,,,,,88@a35d*116@m@aq8z115,95,100
c8.>q6e8q8g z115,,100,88,88q7@b-3415b*3&@b0b*21q8aq7b&
@m64@a15b2@m@az122,,105q8@b-3415g*3&@b0g*21a8
q8o6@u120d*4&@m0,88,,,,,,88@a35e*116@m@aq7z105,92,100
d*40c*16e*16

@u122q7f*3&@m68@a20g*93@a@mz120,85,90,95,100,105,110,115
o6q7r*16g*18>g*14q8{ab<cdef}4

(t2)
@a127o5z110,100egz90,80<c>b<c>b<c>b<c>b<c>b<c>b<c>b<c4@ar4
@a127z75o5'egb<d'z80,90<cdcdcdcdcdcr8@ar2@a127
z110o4az100,90,80,70b-<dfab-<dfa>b-<dfa>b-<dfa4@ar4@a127
o5z110,100fa-z90,80<c>a-<c>a-<c>a-<c>a-<c>a-<c>a-<c>a-@ar2

@a127<c>b<c>b<c>b<c>b<c>b<c>b<c>b<c>b@ar2
@u78o6@a127eeeeeeee eeeeeeee@ar2
@a127z90,80o5afafafaf afafafaf@ar2
o4z110,105,100,95,90,85,80,75@a127
fb-<ce-fb-<ce- e-c>b-fe-c>b-f cfa<cfa<cf gd>bgd>bgd@a

(t3)
q7o4z100,110,98,88'g4<g',0r*18'bd<gb'*14'gb<dg'*16'd2gb'
z85,94,76>gba8<r*18z119,99,82'd-f-a-<c'*14'cegb'*16'c2eg'
z100,110,120,110,100,90,95,105,100,88
<{'d<d''e<e''a<a'}4{'e<e''d<d'>'a<a'}4<{'d<d''e<e''a<a'}4
'd4<d'o4r4z105,95,85,75,110,100
q8'a-<c'<def>q7'a4<cf''d4gb'

z114,100,84r4r*18'g<dgb'*14'a-<e-a-<c'*16'g2<dgb'
z90,78,90,65,85,75,80,90,100'e8gb<d''egb<c''e8gb<d''egb<c'
'egb<e''egb<e'r4gb<c8r*16z100,114,104
q7'da<d'*16'ea<e'*16'da<d'*14>q8

z90,70,80,60,70,50,60,40,50,30,40,20
{'a<da''a<da''a<da''a<da''a<da''a<da''a<da''a<da''a<da''a<d
a''a<da''a<da'}*34
q7z90,100,114,95'a<da'*17<'da<d'*15'ea<e'*16'da<d'*14>q8
z90,70,80,60,70,50,60,40,50,30,40,20
{'a<da''a<da''a<da''a<da''a<da''a<da''a<da''a<da''a<da''a<d
a''a<da''a<da'}*34

o4z100,90,100,110,105,95
q7'g4<df'q8fb-q7<c8>r*2'fa<c'*46'd4gb'

(t4)
o3q7z110'e1b<d'  r1
q8z100,110,90,100,80,90,70,80,60,70,50,60,40,50,30,40
{'b-<f'<a>'b-<f'<a>'b-<f'<a>'b-<f'<a>'b-<f'<a>'b-<f'<a>'b-<
f'<a>'b-<f'<a>}2
z20,30{'b-<f'<a>'b-<f'<a>'b-<f'<a>'b-<f'<a>'b-<f'<a>'b-<f'<
a>'b-<f'<a>'b-<f'<a>}2
o3q7z110,119'a-8<cd'<f8>r4z110,118,108,88q8g<c>a<d>q7r4

@u110'e1b<d'  z115,100,82q8r2r'b<c'<ce>r4
q8z100,110,90,100,80,90,70,80,60,70,50,60,40,50,30,40
{'b-<f'<a>'b-<f'<a>'b-<f'<a>'b-<f'<a>'b-<f'<a>'b-<f'<a>'b-<
f'<a>'b-<f'<a>}2
z20,30{'b-<f'<a>'b-<f'<a>'b-<f'<a>'b-<f'<a>'b-<f'<a>'b-<f'<
a>'b-<f'<a>'b-<f'<a>}2

z110,90,105,88,95,110,118,105o3q7'e-8b-<d'
y64,127q8e-<cr4>y64,0f*3f+*3q7g*18g8f4

(t8)
@u78o4q8'd1egb'  'e1gb'  >'b-1<de'  <'c2.fa-<c''d4fgb'
'd1egb'  'c1egb'  >'b-1<dfa'<  z100,95,88f2'c4fa''d4gb'

(t6,7)
o1q7z110,116,96,120,80,100e4^16b8q5<ebq7eb4.>
z110,95,105,80,80a4..q5aq7a4.q2aaq7
z110,100,92,115,100,88,98b-4^16<q5(e8f),12q7b-<e>b-<q8e8.
@b0,-8191,12q6e8.q7@b0o1q7z110,95,110,115a-4..q5a-q7g4f4

o1q7z110,116,96,120,80,100e4^16b8q5<ebq7eb4.>
z110,95,105,80,80a4..q5aq7a4.q2aaq7
z110,100,92,115,100,110b-4^16<q5(e8f),12q7b-<e>b-<
@b0,-8191,48q6e4.q7@b0>> z90,100,110,100q7e-4<e-4g4f4

(t10)
o3z100,95,108,100,94|:4d+4{d+d+r}4d+4d+4:|
z86,95,108,100,94a4{d+d+r}4d+4d+4
o3z100,95,108,100,94|:2d+4{d+d+r}4d+4d+4:|
z100,95,108,88,95,89,85d+4{d+d+r}4a8d+8{gc+r}4

(t11)
w2z100,80c4g+4c4g+4  c4{g+rc}4z100,80c4g+4  c4g+4c4g+4
c4g+4c4{g+rc}4  z100,80c4g+4c4g+4  c4{g+rc}4z100,80c4g+4
c4g+4c4g+4  c4{g+rc}4@u100c2

/----------------------  F  ----------------------------

(t1,9)
^15o6q8@u127g*3&a*45&a8.&@b0g@u96q8@m50g4^16@mr8.@u122
q8b&<c&>b&<c&>bz120,115,110,105,100,92,100,115,80
agedc>b<cq7d8.q6ez125,,,95,75,115,,,85,60
q8g*2&a*46&a8r*6g*8f*10 g*2&a*46&a8r*6g*8f*10

z122,,118,,115,,120,,105,90q7g*2&a-*14g*2&a-*14g*2&a-*14
q8g*3&a-*21gfz88,86,84,82,80,78{edc>a-gf}4
z75,78,81,84,87,90,93,96,99,102,105,108{rga-b<cd}4
{efgab<c}4>@u119b*3&<@a0,38c*93@a>z110,105b8.a
z100,88,95,100,120,110,106,88,80,120,,100
gegaq6{bge}4@m64q8d4^16@mcg*2&a*10a@u122<q6c8.&>ar8

z110,115,122,125,88,127q6{raa}8<q8c*8q6c*8>q7a*8<q6c8>
z105,112,102q7raa<cz122,,118,,,80,127,110,95,95
q8d*3&e-*45q6d*3&e-*21&dcq8{d>gg&}8@m64g8@mr4

(t2)
@a127o5z110,100egz90,80<c>b<c>b<c>b<c>b<c>b<c>b<c>b<c4@ar4
r*576  o4z110,105,100,95,90,85,80,75@a127cdeg<cdeg
cdeg<cdeg@ar2  r*576

(t3)
o4q7z100,120,115,105,88'd2gb'<'d8g<c''dgb'*42>'dgb'*14
'dg'*16z110,105,92'c8egb''b<g''b4^16<e'
z110,122,108,115,105,95
rq8'b<b'<'c<c'>'b<b''b<b''g<g''e8<e'
r4..q8z80,90,100d*4f*4a*4q7z127,114'b-8.<dea''b-4^16<dea'

z104,118,110,120,90,115,125,85'g4a-<cf'r'g8.a-<cf'
'a-<cfa'*18'a-<cfa'*14'a-<cfa'*16'b<dgb'*18'b<dgb'*14
'b<dgb'*16z110,127,118,108,100,90,80,90,100,110,99
'g4..<cg'q8g*4<c*4g*4d>gdg<c*13d*13g*22q7
z124,94,114,88,110,82r8.'a<e' q8>'c8.g'q7<'eb'>
q8'ce8.'q7'gb' >'b4<e'z100,105,110,115,120,127

o4'a<d'*16'a<e'*16'a<a'*16<'d<d'*16'e<e'*16'a<a'*10q8
z80,100,70,90,60,80,50,70,40,60,30,50,20,40,10,30
```

```
{a<a>a<a>a<a>a<a>a<a>a<a>a<a>a<a>}4^*6r@u116q7'e8.<e'
z110,100,122,115'c4<c'rcd8c2

(t4)
q7o3z110,120,90,80e2<g+4>r*18g+*14g+*16r2r
z110,120,110,110,100,90'g<ce''g<ce''g<ce''g<ce''g<ce'
'g8b<c'  z112,105'b-2.<de'r'b-8.<de'

z95,110,90,80,110,100q8g*3q7'a-<cf'*93q8f*2f+*2q7g*44f4
z110'el<c'  z115,95r4..'ab<e'r8.'a4^16b<e'

z100,110,125'b-<f'*18'b-<f'*14'b-<f'*16q8
z90,110,80,100,70,90,60,80,50,70,40,60,30,50,20,40
{'b-<f'<a>'b-<f'<a>'b-<f'<a>'b-<f'<a>'b-<f'<a>'b-<f'<a>'b-<f'<a>'b-<f'<a>}4
q7z110,92'b-8.<fa'<d>q8
z90,110,80,100,70,90,60,80,50,70,40,60,30,50,20,40
{'b-<f'<a>'b-<f'<a>'b-<f'<a>'b-<f'<a>'b-<f'<a>'b-<f'<a>'b-<f'<a>'b-<f'<a>}4
z98,88,120,100f*2b-*2<'cfb-'*92>'g+2<dg',3

(t8)
@u80q8o4'elgb'  'elgb'  'fla'  z100,109,95,100,95
f2<c4>b4  <cl>  bl  z80,65,75al  'c2fb-''d2g<c'

(t6,7)
o2q6z110,90,100,110,95e4.q7>b8<c8.e8.q6g8
z110,100,80,110,100,80q6a4r8q7eq2eq6>a4.q7<e>q2a
q6z110,90,100,110,95b-4.q7<c8d8.e8.q6c8>

z110,118,100,110,100q7a-4<q6e-8>q7a-8g4f4
o3z110,,115,110,105,100,95,90,110,100,90q7
@b-3415,0,1e*4&@b0e*56q6(e8d),12q7c>bage{c>bg}4
z110,118,100,110,100q7a*60q6(b8<c),12q5eq7g4>a4

z110,100,80,110q7b-4.<fq2fq6>b-2 z110,100<e-2>e2

(t10)
z100,95,108,100,94o3|:2d+4{d+d+r}4d+4d+4:|
z100,95,108,100,94,108d+4{d+d+r}4d+4{d+d+r}4
z100,95,108,100,94d+4{d+d+r}4d+4d+4
z88,95,108,100,94c+4{d+d+r}4d+4d+4
z100,95,108,100,94|:2d+4{d+d+r}4d+4d+4:|
z100,95,108,80,,75,85d+4{d+d+r}4d+8d+8{agr}4

(t11)
z100,80|:3c4g+4c4g+4:|  c4g+4c4{g+rc}4
z100,80|:4c4g+4c4g+4:|

/----------------------  G  ------------------------------

(t1,9)
q8_10z115,102,122,,98o6e8a8b*1&<c*23>a@u110a&b4.@u98q7a8

z105,100,98,92,,102q7g4q8f8eq8e&f4@ar8q7g8
z119,,110,98,100,110,124,,,110,110
q8f*3&g*45f8q7e8q8f8a-8<d*3&e*45&q4@a59e2@a-94@m64d2&
¥75q8d1@m

(t2)
@a127o4r2r8z70,80,90,100,110cef+ab4@a@a127r8
z110,100,90,80dfa<c>fa<c16>@ar8@a127
z120,115,110,105,100,95,90,85,90,95,100fa<cdfa<cdc>af16@a
r8o3@a127z120,115,110,105,100,95,90,85,80,75,70,65,60
ga<cdfga<cdfga<c8>>efa-b<efa-b<e4

(t3)
q7o4z110,120,127,108,118,125,95'c8e''c8ea''c8a<c''c8ea'
'c4ef+b''e8f+b''c8ef+a'>z125,110'a4<db''a4<df'<
z110,118,100,80'c4df''da<c'gc8z115,125,115,120,125,127
o4'c4dg''cdf8''ce8'>'b<df8'<'dfa-8't-10'f4a-<e',3
@d100z105,95,80,65t-20r2r.q8a*6f+*6e*6>b*6t+8o2
z70,75,80,85,90,95{c<ct+2b<e t+2fb}4z88,95,100,105

{fat+4b<e z88,105,120,80,95,110,70,85,100et-5f+t-5bt-5at-5b<e>t-5bt-5<et-5b}2
@u79t-3f+2

(t4)
o2z110,118,108,98,88q7'a2<e'r8q8<c*6e*6f+*6<c*6>>q7r4
z110,100,112'a2.<f'r8g8  'g1<g'

(t5)
q8z105,92,112,88o4e8a8<c8>a@u100ab4.@u88a8z95,90,88,78,,62
g4f8eef4r8g8z109,100,88,90,100,108,,90,90g4f8e8f8a-8<e4&
q5e2q8>@a-60e2&  ¥75e1

(t6,7)z110,95q7a2...q6aq7  z110,115,80,120,100,90
a2a8.q2aq7<ae>q7a8  z110,80,100q7g2^8.q2gq7g4  @u104q8c1&
¥75c1

(t8)
z85,80q8o4'e2a<c''e2ab'  @u80'f1a<c'
r2.@u75<'d4b'&  'd4b'r2.  r1

(t10)
z88,100,105,92c+4d+4d+4d+4  z95,100,105,92d+4d+4d+4d+4
z95,108,90,100,110,120d+4d+8d+8d+8d+8d+4

(t11)
w5z100,80,80|:2c4g+2g+4:|  c1  r1r1

(p)
```

リスト9 TIME STREAM用カウンタ表示

```
 1:00003540 00000000   2:00001440 00000000   3:0000353A 00000000   4:000033C0 00000000
 5:000003C0 00000000   6:00003540 00000000   7:00003540 00000000   8:00002340 00000000
 9:00003540 00000000  10:000033C0 00000000  11:00003540 00000000
```

(善)の「勝負はこれからだ」

去年買ったSONYの29インチモニタKX-29HV3で最近画面がやたら左によってしまうようになった。ブラウン管の右枠付近2cmは電子ビームが当たらないほど表示が左に寄ってしまっている。このモニタはRGB入力端子1系統，Sビデオ入力，ビデオコンポジット入力がそれぞれ5系統，そしてスピーカー出力端子，モニタ出力端子まで装備と，かなり贅沢な作りをしており，画質もかなりよい。RGB入力端子付きなのでゲーム機もOK，スキャンコンバータを通してだがX68000の画面なども出力させたりすることもできる。ビデオやLDを見たりするときにも使える。入力端子が多いので背面の配線替えなんかもほとんどしなくて済むし，大変便利なのだが，たったひとつものすごい欠点を持っている。

それは垂直振幅/水平振幅/垂直位置/水平位置といったモニタでは当たり前の端子が一切ついていないのである。だから画面がただ左によってしまっただけでサービスマンを呼ばなくてはならないことになるのだ。このままでは使いづらいのでしかたなくサービスマンを呼んだ。

サービスマン「これですか……確かに左によっちゃってますね。じゃ右にずらしましょう」

背面パネルとか開けるのかね，といろいろ想像していたら，

サービスマン「お客さん，リモコンありますか，このモニタの」

私「???」

いわれたとおりリモコンを渡す。リモコンを受け取るとそのサービスマンはリモコンのスイッチを押しはじめた。まるで格闘ゲームの隠れキャラ使用コマンドをいれるように。……と，なんと，モニタは設定モードになったではないか！ 垂直振幅/水平振幅/垂直位置/水平位置はもちろん30近いさまざまなパラメータをリモコンのキー操作で変更できるようだ。

「へぇ，こりゃ便利ですね。これ教えてくださいよ。でも，なんでこんな便利な機能マニュアルに載せないんですか」

「変に設定しちゃったときに故障と間違われると困るからでしょう。あ，別にいいですよ，秘密事項でもなんでもないですから」

うーんメーカーは消費者を完全にバカと思っているな……。で，このサービスマンが教えてくれた各種設定モードへ入るためのコマンド，ちょっとここで教えちゃおう(KX29HV3用)。

【設定モードへ】［電源(OFF)］→［画面表示］→［5］→［音量＋］→［電源(ON)］

【設定モードでのキー操作】

項目切り換え［1］［4］ パラメータ変更［3］［6］

設定の書き込み［ミュート］→［12］

データ読み込み［10］→［12］

標準データに戻す［8］→［12］

【設定モード解除】［電源OFF］

身の周りのほかの家電品にもこういう隠れコマンドがあるのかしらん。ちょっとわくわくしてきたよ。

バックナンバー案内

BACK ISSUES

ここには1994年10月号から1995年9月号までをご紹介しました。現在1994年8～12月号，1995年4～6，9月号の在庫がございます。バックナンバーはお近くの書店にご注文ください。定期購読の申し込み方法は128ページを参照してください。

1994



10月号
特別企画　もみじ狩りPRO-68K
連載　響子in CGわ～るど/ショートプロ/ハードコア3D
TeX入門講座/ゲーム作りのKNOW HOW/善バビ
猫とコンピュータ/ファイル共有の実験と実践
●特別付録　もみじ狩りPRO-68K(5″2HD)
●新製品紹介　F-Card V5 for x68k
LIVE in '94　イース2/MSX用GRADIUS2/NATURE
THE SOFTOUCH　スーパーストII/スターラスター　他
全機種共通システム　怪しいZ80の使い方/ゲーム作成講座(3)



11月号
特集　STEP UP BASIC
連載　響子 in CGわ～るど/ショートプロ/ハードコア3D
TeX入門講座/DōGA CGアニメーション講座
システムX探偵事務所/ローテク工作/善バビ
●新製品紹介　BJC-400J/X680x0 Develop. & libcII
Free Software Selection Vol.2
LIVE in '94　ダーク・スペース/ENDLESS RAIN/レナのテーマ
THE SOFTOUCH　スーパーストII/餓狼伝説SPECIAL
全機種共通システム　B-GALETS2



12月号
特別企画　XL/Imageお試し版＋α
連載　響子in CGわ～るど/ショートプロ/ハードコア3D
ファイル共有の実験と実践/DōGA CGアニメーション講座
システムX探偵事務所/ローテク工作/TeX入門講座
●特別付録　XL/Imageお試し版＋α(5″2HD)
●新製品紹介　H.A.R.P/XDTP SX-68K
LIVE in '94　幻想即興曲/きまぐれ　オレンジ☆ロード　他
THE SOFTOUCH　魔法大作戦/スーパーストII
全機種共通システム　シューティングゲーム作成講座(4)

1995



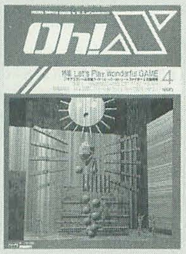
1月号(品切れ)
特集　割り切って使うCD-ROM
連載　響子 in CGわ～るど/ショートプロ/ハードコア3D
ファイル共有の実験と実践/DōGA CGアニメーション講座
システムX探偵事務所/ローテク工作/TeX入門講座
●CD-ROMドライブ紹介　CS-CD301X/CDS-E/SCD-200
●新製品紹介　X68000XVI用アクセラレータXellent30
LIVE in '95　ぷよぷよ/ジムノペディNO.1/PRIME
THE SOFTOUCH　バックランド/上海 万里の長城/魔法大作戦
餓狼伝説SP 特別編/スーパーストII 特別編



2月号(品切れ)
特集　MicroProcessingUnit
連載　響子 in CGわ～るど/ショートプロ/ハードコア3D
SX-BASIC公開デバッグ/DōGA CGアニメーション講座
システムX探偵事務所/SX-WINDOWによるDTP
●特別企画　最新ゲーム機を見る
●新製品紹介　Datacalc SX-68K/シャーペンワープロパック
●1994年度GAME OF THE YEARノミネート作品発表
LIVE in '95　サムライスピリッツ/AFTER SCHOOL/白鳥の湖
THE SOFTOUCH　スーパーストII 特別編



3月号(品切れ)
特集　SoundEffects
連載　響子 in CGわ～るど/ショートプロ/ハードコア3D
システムX探偵事務所/ファイル共有の実験と実践
ピコピコエンジン活用講座/SX-WINDOWによるDTP
●SX-WINDOW用ユーティリティ　どっち.X
LIVE in '95　魔法のプリンセスミンキーモモ/別れの曲
ファイナルファンタジーII/宇宙戦艦ヤマト完結編
THE SOFTOUCH　ディグダグ/ディグダグII/VIEW POINT
全機種共通システム　S-OSシステムコールライブラリ



4月号
特集　Let's Play Wonderful GAME
連載　響子 in CGわ～るど/ショートプロ/ハードコア3D
システムX探偵事務所/ファイル共有の実験と実践
DōGA CGアニメーション講座/ローテク工作
●1994年度GAME OF THE YEAR発表
●新製品紹介　TS-6BS1mkII/MJ-5000C/MATIER ver.2.1
LIVE in '95　天聖龍/ファイナルファンタジーVI/
ANOTHER DAY/ハートオブザマッドネス
全機種共通システム　S-OSねちねち入門(1)



5月号
特集　Realize Graphic
連載　響子 in CGわ～るど/ショートプロぱーてぃ
ローテク工作実験室/SX-BASIC公開デバッグ
システムX探偵事務所/ANOTHER CG WORLD
●特別付録　Oh!電脳倶楽部
●新製品紹介　フォント＆ロゴデザインツール
LIVE in '95　ドラゴンセイバー/ミッドナイトレジスタンス 他
THE SOFTOUCH　ボンバーマン ぱにっくボンバー
全機種共通システム　S-OSねちねち入門(2)



6月号
特集　Open the SX-WINDOW
連載　響子 in CGわ～るど/ハードコア3Dエクスタシー
DōGA CGアニメーション講座/ローテク工作実験室
システムX探偵事務所/ショートプロぱーてぃ
●特別企画　X68000周辺機器パワーアップ計画
●新製品紹介　Xellent30s/学研統合電子辞書 for SX-Window
●第6回アンケート分析大会
LIVE in '95　クリティカルポイント/THE SUMMER OF '68 他
全機種共通システム　S-OSねちねち入門(3)/BLOCK DOWN



7月号(品切れ)
特集　Optimizing Method
連載　響子 in CGわ～るど/ハードコア3D/ファイル共有
DōGA CGアニメーション講座/ショートプロぱーてぃ
システムX探偵事務所/ANOTHER CG WORLD
●THE USER'S WORKS SPECIAL
●新製品紹介　PDドライブLF-1000
THE SOFTOUCH　バラデューク
LIVE in '95　クロノ・トリガー/SUPER MARIO BGM集 他
全機種共通システム　FE ver.1.0



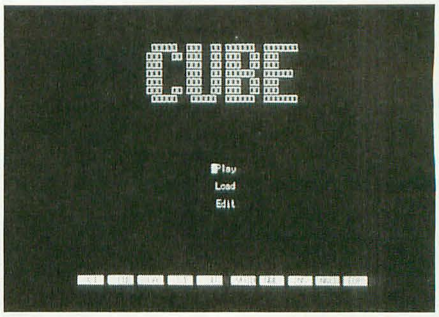
8月号(品切れ)
特別企画　暑中見舞いPRO-68K
連載　響子 in CGわ～るど/(善)のゲームミュージック
DōGA CGアニメーション講座/ショートプロぱーてぃ
システムX探偵事務所/ANOTHER CG WORLD
●特別付録　暑中見舞いPRO-68K(5″2HD)
●新製品紹介　SCSI2ボードMach-2/DSPボードAWESOME-X
CD-ROMドライブCDG-TX4
LIVE in '95　淡紅色の夢/Tomorrow never knows　他
全機種共通システム　IF ONLY



9月号
特集　Animation Now!
連載　響子 in CGわ～るど/ハードコア3D/DSP
DōGA CGアニメーション講座/ショートプロぱーてぃ
システムX探偵事務所/ローテク工作実験室
●音声波形表示プログラム　OCR.X
LIVE in '95　ファイナルファンタジーV/SAY ANYTHING
ときめきメモリアル/ドラゴンスレイヤーVI　他
全機種共通システム　FE ver.1.0ラインプリントルーチン詳細
MISSILE SYSTEM

THE SENTINEL

〈対応機種一覧〉●MZ-80K/C/700/1500●MZ-80B/2000●MZ-2500/2861●X1●X1turbo/Z●PC-8001/8801/88●SMC-777/C●PASOPIA/5●PASOPIA/7●FM-7/77/AV●MSX/2/2+/turbo R●PC-286/386/486/9801/98/9821●X68000/X68030
掲載されたプログラムの利用には各機種用のS-OS"SWORD"システムが必要です。



第162部　パズルゲームCUBE

●キューブで挟め！

今月はSLANG用パズルゲーム「CUBE」が登場です。ゲームの目的は，5種類あるキューブのうち画面にあるマッドキューブを全部消すこと。キューブの消し方は，同じ種類のキューブで，別の種類のキューブを挟み込むだけです。ただし，消せるキューブは1種類のみ。うまくキューブを挟み込んでください。

ちなみにこのゲームでは，敵キャラクターなどはいませんので，リアルタイムアクションが苦手な人も大丈夫。じっくり考えながらパズルを解いてみてください。

そして，面エディタも標準でついていますので，用意された30面を解いてしまったら，今度はオリジナル面を作成して楽しみましょう。というようなことを書くと次にくる言葉は決まっていますね。そうです。ここで，「CUBE」オリジナル面の募集をしちゃいます。

宛先はいつものとおりで，

Oh!X編集部

「CUBEオリジナル面データ」係

まで，よろしくお願いします。

さて，面データを作成していただくのはいいとして，問題となるのはデータサイズです。掲載された面データのダンプリストを見ていただくとわかりますが，結構サイズは大きいものです。複雑なデータではないので，入力する手間はそれほどかからないと思われますが，面数が増えれば増えるほど誌面での掲載が難しくなってしまいます。ということで，どなたか汎用性はなくてもかまいませんから，簡易圧縮ツールを作りませんか？　こちらのほうも投稿をお待ちしています。

また，現在X68000版への移植も順調に進んでます。こちらも掲載予定がありますので，X68000ユーザーの皆さんは，楽しみにしていてください。ちなみに，面データは共通となりますので，X68000版に発表されると思われる面データをS-OS版でも使用することができるでしょう。S-OSユーザーの方も楽しみにしていてください。

パスルゲームついで(?)ということで，もうひとつ嬉しい情報があります。それはS-OS版「PICTパズル」の投稿がありました。こちらのほうは現在チェック中です。なにも問題がなければ，近いうちに掲載することができるでしょう。

●「BREEZE」完成！

以前，このTHE SENTINELで紹介したPCM再生プログラム「BREEZE」の完成版が届きました。X1,MZ-80/1200/700/1500/80B/2000/2200/2500,PC-8801SR以降,MSXのソースリスト共通化作業が終わり，動作チェックがすめば発表可能ということです。このソースリスト共通化作業は，かなり苦労した様子でほとんど全面ソースリストを書き直したものもあるとか。そのかいあって，完成したのものは，このTHE SENTINELのコーナーでも掲載可能な大きさにまとまっています。

そして，ネックとされていたデータの作成(各種PCMデータのコンバータ)とメモリを有効利用するメモリ管理ドライバの制作も進められているようです(「BREEZE」との同時発表は難しそうとのこと)。

オブジェクト共通化作業以外にも仕様決定段階で「すべての機種で動く」ことを前提としたため，各機種固有の機能を切り捨てたりと苦労の多かったこの作品。8ビット機でもここまでできるんだ，ということを見せつけてくれる，いい手本となるでしょう。

ちなみに投稿原稿には，データの上位互換をもつ次期PCM再生プログラム「BLAST(仮)」の企画もあるとかないとか……。機種が限定されてしまうと，THE SENTINELでは扱うことが難しくなってしまいますが，自分で納得のいくまでがんばってみてください。

●今月のフリーソフト

最後に今月のフリーソフトです。今月は8月号で掲載した「IF ONLY」の制作者，森沢さんから「フリーソフト化してもいいですよ」との連絡をいただきました。森沢さん，ご協力感謝します。

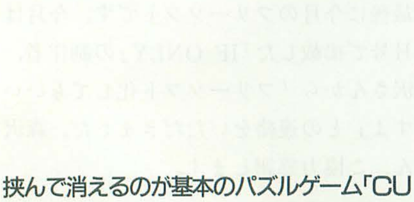
1995■インデックス

全機種共通
S-OS"SWORD"要

パズルゲーム CUBE（要SLANG）

Itou Masahiko
伊藤 雅彦

挟んで消えるのが基本のパズルゲーム「CUBE」。4種類あるキューブを使って，マッドキューブを消すのだ。ちょっとボリュームがあるけどがんばって打ち込んで楽しんじゃいましょう。



今回発表する「CUBE」は，SLANG用の単純明快パズルゲームです。簡単すぎず，難しすぎず手頃な難易度なので，ちょっと頭を捻ってみてください。

入力&起動方法

まず，テキストエディタでソースリスト（リスト1）を入力し，セーブします。そしてSLANGを立ち上げて，

]Cソースファイル名

と打ち込むと，アドレス8400_H以降にオブジェクトが生成されます。このオブジェクトは3000_H以降にロードして実行すべきものになっていますから，

]S実行ファイル名:3000:xxxx:3000:8400

と打ち込んでセーブしてください（xxxxはコンパイル時に"Program 3000 - xxxx"と表示された値）。

続いて，面データ（リスト2，3）をMACINTO-Cなどを使って入力し（アドレスはどこからでもかまいません），ファイル名を必ず"CUBE.CBS"としてセーブしてください。以上で入力は完了です。

プログラムを起動するときには，実行ファイルをロードして，面データのファイル"CUBE.CBS"のあるデバイスをデフォルトデバイスにしてから，

#J3000

と打ち込んでください。"CUBE.CBS"をロードしたあと，タイトル画面が表示されるはずです。

ちなみに，テープにも対応しているはずです。"CUBE.CBS"をセーブしたテープをセットしてプログラムを起動してください。そして，コンパイル時には全ソースをメモリに置く余裕などありませんから，ソースをいくつかに分割したうえで#CHAINコマンドをつけ加える手を使ってください。

ルール説明

フィールド上にキューブとマイキャラが置いてあります。マイキャラは上下左右に動くことができ，またキューブを一度にひとつだけ横方向に押すことができます。そしてキューブは，下になにもないと落下していきます。

キューブにはノーマルキューブ，3種類のパワードキューブ，マッドキューブがあります（図1）。パワードキューブとマッドキューブは，ほかの種類のキューブを縦，横に挟むことによってそのキューブを消すことができます。挟まれるキューブは，同じ種類ならいくつであってもかまいません（図2）。

で，キューブをうまく動かして，画面上のマッドキューブをすべて消すと面クリアとなります。ルールはこれだけ，単純なものでしょう。

遊び方

まず，使用するキーですが，基本的に上下左右と決定，解除の6種類です。上下左右はテンキー，カーソルキー，I・M・J・Lキーが使えます。決定はリターンキー，スペースキー，Xキーで，解除はSHIFT＋BREAK，CLRキー，Qキーです。

プログラムを実行すると，タイトル画面が出て"Play"，"Load"，"Edit"の機能選択になります。ここで"Play"を選んで決定キーを押すと，面選択になります。用意された面は15×2＝30面です。好きな面を選んで決定してください。これでゲームが始まります。

ゲーム中は方向キーでマイキャラを動かします。マイキャラは不死身でタイムアップなどもないので，ギブアップしたいときには解除キー（SHIFT＋BREAK）を押してください。面選択に戻ります。ゲームを終えたいときには面選択で解除キーを押して，タイトル画面になったところでさらに解除キーを押します。

エディット機能

このゲームは，全30面をクリアしたら終わりじゃありません。オリジナル面を作ってみましょう。タイトル画面で"Edit"を選択すると，エディタに入ります。

まず，エディタの機能選択メニューが出ます。"Rnd.Edit"（Round Edit），"Stg.Edit"（Stage Edit），"Load"，"Save"の4つがあります。順番に説明しましょう。

●Round Edit

面の内容を編集します。"Field"，"Cube"，

“Round”のメニューから機能を選択します。

a)　Field

地形を作ります。カーソルを方向キーで動かして，決定キーを押すと地形の有無が反転します。

b)　Cube

キューブを配置します。まず置きたいキューブを選んでから，置きたい場所にカーソルを動かして決定キーを押してください。すでにキューブが置いてある場所で決定キーを押すと，そのキューブが消え，消したキューブが次に置くキューブとなります。また，マイキャラの配置もキューブと同様にこの機能で行います。

c)　Round

編集する面を変更します。

編集中の面にひとつもキューブがない場合，マイキャラがないか2つ以上ある場合には，面を変更したりRound Editを抜けたりすることはできません。注意してください。

●Stage Edit

このゲームでは，いくつかの面をまとめてひとつの面データファイルにしたものをステージと呼ぶことにします。Stage Editでは，面の順番の入れ替えや新しい面の挿入，面の削除を行います。“Move”，“Insert”，“Delete”，“Title”のメニューから機能を選択します。

a)　Move

指定した面を別の場所に移動します。たとえば3面を6面に移動したいときには，カーソルを3面に合わせて決定キーを押し，続いてカーソルを6面のところに合わせて決定キーを押します。

b)　Insert

新しい面を挿入します。挿入したい場所にカーソルを合わせて決定キーを押してください。メモリが足りなくて挿入できないときには，“Limit”というメッセージが表示されます。

c)　Delete

指定した範囲の面を削除します。たとえば2面から5面を削除するときは，カーソルを2面に合わせて決定，続いて5面に合わせて決定します。

まったく新しいステージを作るときには，すべての面を削除してから新しい面を挿入し，面を作成していってください。

d)　Title

ステージ名を入力します。今回掲載した面データでは“Mercury Stage”となっています。最初に作ったステージだからという理由で，SEGA SATURNの命名方法をパクってつけました。こんなふうに，自作のステージにお好きなように名前をつけてください。8桁×2行です。

入力時にはカーソルキーの左右がカーソル移動，上がバックスペース，下がスペース挿入で，クリアキーが全消去になっています。リターンキーで入力確定，SHIFT＋BREAKで取り消しになります。

図1　キャラクタ紹介

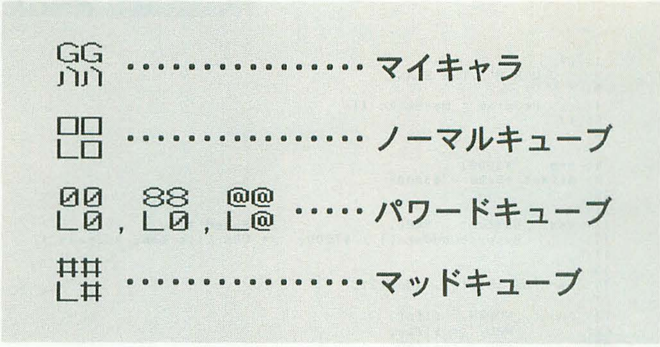
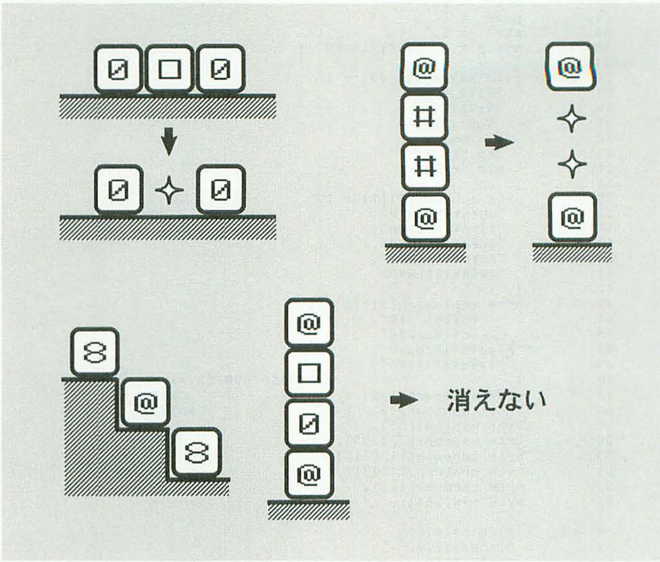


図2　キューブの消し方



●Load

面データをロードします。ファイル名を入力してください。入力時にはステージ名入力と同様のキー機能が使えます。

●Save

面データをセーブします。ロード時と同様に，ファイル名を入力してください。拡張子は自動的に“CBS”になります。

ゲーム起動後，以前にセーブした面データでプレイするときには，タイトル画面かエディタのメニューで“Load”を選んで，面データをロードしてからゲームを始めてください。

そのほかの注意事項

ゲームスピードは，ソースリストの冒頭にある“waitcnt＝6000”のところを変更して調節してください。また，独自の関数をSLANGに追加している方は，オブジェクトのサイズが大きくなりすぎて正常に実行できない場合があるかもしれませんので，注意してください。サイズが$3FF0_H$バイトぐらいより大きいと，プログラム/動的変数の領域と面データのロード領域が重なってしまいます。この場合，ソースリスト冒頭の“rounddata[] ＝$7000”の部分を書き換えて，面データのロード領域を後ろにずらしてください。もっとも，関数を追加しないSLANGでコンパイルするにこしたことはありませんが（NEWファイル入出力関数を追加しただけのものだったら，サイズの心配は無用です）。

それから，1ステージ最大255面まで入れられるようになってはいますが，データ量が61,727バイトにもなってしまうので，そんな大きな面データをロードすることはどんな機種でも不可能です（エディタでは作成することもできませんが）。100面ぐらいなら，どの機種でもロードできるでしょう。自作の面を多くの人に楽しんでもらおうと思うなら，1ステージ100面までに抑えておいてください。

では，土壇場になって面データを作成してくれた高橋氏に感謝しつつ，ごゆっくり楽しんでください。

リスト1 CUBE.SLA

```
/*
  C U B E   for SLANG

     Program : Masahiko Ito
*/


org    $3000;
offset $8400 - $3000;


var   waitcnt = 6000,              /* Speed */
      byte rounddata[] = $7000;   /* CBS File Load Address */


machine ldir(3), lddr(3);

const _WOPEN = $1faf,
      _WRD   = $1fac,
      _RDD   = $1fa6,
      _FILE  = $1fa3,
      _ROPEN = $2009,
      posx  = 0,
      posy  = 1,
      atr   = 2,
      myshp = ["GG¥d¥l¥lハハ¥0"];

array byte keyassign[23] = [
         "¥l4Jj",
         "¥r6Ll",
         "¥u8Ii",
         "¥d2Mm",
         "¥n Xx",
         $1b,"¥cQq"
      ],
      byte cubeshp[4][7] = [
         "ロロ¥d¥l¥lLロ¥0",
         "##¥d¥l¥lL#¥0",
         "00¥d¥l¥lL0¥0",
         "88¥d¥l¥lL8¥0",
         "@@¥d¥l¥lL@¥0"
      ],
      byte explosion[3][7] = [
         "  ¥d¥l¥l  ¥0",
         "..¥d¥l¥l..¥0",
         "oo¥d¥l¥loo¥0",
         "**¥d¥l¥l**¥0"
      ],                          /* 1234567890123 */
      byte accesscbs[17] = ["CUBE         .CBS¥0"],
      byte inputbuf[13]  = ["            ¥0"],
      byte map[23][23],
      byte mapcopy[23][23],
      byte cubestat[120][2],
      byte cbstcopy[120][2],
      byte chkcube[121],
      byte eda[254];

var   _SIZE :$1f72,
      _DTADR:$1f70,
      _EXADR:$1f6e,
      _MEMAX:$1f6a,
      byte rnddt[][10],
      maxround,
      round,
      cubes,
      mads, pmads,
      myposx, myposy,
      edx, edy, edi, edj;


main()
  begin

  width(40);
  print("¥c");
  if (loadcbs() == false) [
    loop [
      case (title()) [
        1: play();
        2: load();
        3: edit();
        others: exit;
      ]
    ]
  ]
  print("¥c");

  end;


title()
  array byte logo[6][3] = [
           $7c,$e6,$fc,$fe,
           $e6,$e6,$e6,$e0,
           $e0,$e6,$e6,$e0,
           $e0,$e6,$fc,$fc,
           $e0,$e6,$e6,$e0,
           $e6,$e6,$e6,$e0,
           $7c,$7c,$fc,$fe
         ];

  var i, j, k, l;

  begin

  print("¥c");

  for i=0 to 6 [
    locate(4,4+i);
    for j=0 to 3 [
      k = logo[i][j];
      for l=7 downto 0 [
        print(str$(bit(k,l) ? 'ロ' : ' ',1));
      ]
    ]
  ]

  locate(18,18);
  print("Play");
  locate(18,20);
  print("Load");
  locate(18,22);
  print("Edit");

  i = 0;
  loop [
    locate(17,i*2+18);
    case (inkey1()) [
      2: if (--i == -1)  i = 2;
      3: if (++i ==  3)  i = 0;
      4: exit;
      5: return(0);
    ]
  ]

  end(i+1);


play()
  begin

  initscr();
  locate(29,22);
  print(!(&cubeshp[1][0])," =");
  while (selectround() == false)  game();

  end;


game()
  begin

  loop [
    initwork();
    makescr();
    locate(34,23);
    print(form$(mads,3));
    chkers_s();
    if gamemain()  exit;
    if (++round > maxround)  round = 1;
  ]

  end;


gamemain()
  var finctr, i;

  begin

  finctr = 5;

  repeat [
    if ((i = moveme()) == $ffff) return(true);
    fallexpld(i);
    wait();
    chkers_f();
    prtmads();
    if (mads == 0)  finctr--;
  ] until (finctr == 0);

  locate(8,12);
  print(" C L E A R ! ¥l");
  inkey(1);

  end(false);


moveme()
  var rc, i, j;

  begin

  rc = $ff;

  i = inkey(0);
  for j=0 to 23 [
    if (i == keyassign[j])  exit;
  ]
  case (j >> 2) [
    0:[
      if (map[myposy][myposx-1] == $79) [
        if (map[myposy+1][myposx-1] == $79) [
          map[myposy  ][myposx-1] =
          map[myposy+1][myposx-1] = $7b;
          map[myposy  ][myposx+1] =
          map[myposy+1][myposx+1] = $79;
          locate(--myposx+2,myposy);
          print(!(myshp)," ¥u¥l ");
        ]
      ] else [
        if (myposx > 3) [
          if ((i = map[myposy][myposx-2]) >= $80) [
            if (map[myposy][myposx-3] == $79) [
              if (map[myposy+1][myposx-3] == $79) [
                map[myposy][myposx-3] = i;
                map[myposy  ][myposx-2] =
                map[myposy+1][myposx-3] = i and $7f;
                map[myposy  ][myposx-1] =
                map[myposy+1][myposx-1] = $79;
                rc = i = i and $7f;
                locate(--cubestat[i][posx]+2,myposy);
                print(!(&cubeshp[cubestat[i][atr]][0]),
                        " ¥u¥l ");
              ]
            ]
          ]
        ]
      ]
    ]
    1:[
      if ((i = map[myposy][myposx+2]) == $79) [
```

```
        if (map[myposy+1][myposx+2] == $79) [
          map[myposy  ][myposx+2] =
          map[myposy+1][myposx+2] = $7b;
          map[myposy  ][myposx] =
          map[myposy+1][myposx] = $79;
          locate(++myposx+1,myposy);
          print(" ¥d¥l ¥u",!(myshp));
        ]
      ] else [
        if (myposx < 20) [
          if (i >= $80) [
            if (map[myposy][myposx+4] == $79) [
              if (map[myposy+1][myposx+4] == $79) [
                map[myposy][myposx+3] = i;
                map[myposy  ][myposx+4] =
                map[myposy+1][myposx+4] = i and $7f;
                map[myposy  ][myposx+2] =
                map[myposy+1][myposx+2] = $79;
                rc = i = i and $7f;
                locate(++cubestat[i][posx]+1,myposy);
                print(" ¥d¥l ¥u",
                      !(&cubeshp[cubestat[i][atr]][0]));
              ]
            ]
          ]
        ]
      ]
    ]
    2:[
      if (memw[&map[myposy-1][myposx]] == $7979) [
        memw[&map[--myposy][myposx]] = $7b7b;
        memw[&map[myposy+2][myposx]] = $7979;
        locate(myposx+2,myposy);
        print(!(myshp),"¥d¥l¥l  ");
      ]
    ]
    3:[
      if (memw[&map[myposy+2][myposx]] == $7979) [
        memw[&map[myposy  ][myposx]] = $7979;
        memw[&map[myposy+2][myposx]] = $7b7b;
        locate(myposx+2,myposy++);
        print("  ¥d¥l¥l",!(myshp));
      ]
    ]
    5: rc = $ffff;
  ]

  end(rc);


chkers_s()
  var i, cb;

  begin

  ldir(map,mapcopy,24*24);
  ldir(cubestat,cbstcopy,cubes+cubes+cubes);

  for cb=cubes-1 downto 0 [
    chkerase(cb, 0, 2);
    chkerase(cb, 2, 0);
  ]

  end;


chkers_f()
  var cci, cb;

  begin

  ldir(map,mapcopy,24*24);
  ldir(cubestat,cbstcopy,cubes+cubes+cubes);

  cci = 0;
  while ((cb = chkcube[cci++]) != $ff) [
    chkerase(cb, 0, 2);
    chkerase(cb, 2, 0);
  ]

  end;


chkerase(cb, dx, dy)
  array byte sec[11], byte oec[11];

  var x1, y1, x2, y2, a, seci, oeci, oa, osw,
      i, j, k;

  begin

  oa = osw = seci = oeci = 0;
  a = cbstcopy[cb][atr];

  x1 = (x2 = cbstcopy[cb][posx]) - dx;
  y1 = (y2 = cbstcopy[cb][posy]) - dy;
  if (((x1 or y1) and $ff00) == 0) [
    if ((i = mapcopy[y1][x1]) >= $80) [
      if ((j = cbstcopy[i = i and $7f][atr]) == a) [
        oec[oeci++] = cb;
        loop [
          repeat [
            oec[oeci++] = i;
            if (((x1 = x1 - dx) or (y1 = y1 - dy)) and $ff00) [
              exit(2);
            ]
            if ((i = mapcopy[y1][x1]) < $80)  exit(2);
          ] until ((j = cbstcopy[i = i and $7f][atr]) != a);
          oa = j;
          exit;
        ]
      ] else [
        oec[oeci++] = cb;
        oa = j;
        if (a) [
          k = $ffff;
          repeat [
            sec[seci++] = i;
            if (((x1 = x1 - dx) or (y1 = y1 - dy)) and $ff00) [
              exit;
            ]
            if ((i = mapcopy[y1][x1]) < $80)  exit;
          ] until ((k = cbstcopy[i = i and $7f][atr]) != j);
          if (k != a)  seci = 0;
        ]
      ]
    ]
  ]

  sec[seci] = $ff;
  erase(sec);

  seci = 0;
  if ((((x2 = x2 + dx) or (y2 = y2 + dy)) and $ff00) == 0) [
    if ((i = mapcopy[y2][x2]) >= $80) [
      if ((j = cbstcopy[i = i and $7f][atr]) == a) [
        if (oa) [
          loop [
            repeat [
              oec[oeci++] = i;
              if (((x2 = x2 + dx) or (y2 = y2 + dy))
                  and $ff00) [
                exit(2);
              ]
              if ((i = mapcopy[y2][x2]) < $80)  exit(2);
            ] until ((j = cbstcopy[i = i and $7f][atr]) != a);
            if (j == oa)  osw = true;
            exit;
          ]
        ]
      ] else [
        if (j == oa)  osw = true;
        if (a) [
          k = $ffff;
          repeat [
            sec[seci++] = i;
            if (((x2 = x2 + dx) or (y2 = y2 + dy)) and $ff00) [
              exit;
            ]
            if ((i = mapcopy[y2][x2]) < $80)  exit;
          ] until ((k = cbstcopy[i = i and $7f][atr]) != j);
          if (k != a)  seci = 0;
        ]
      ]
    ]
  ]

  sec[seci] = $ff;
  erase(sec);
  if (osw)  if (oa) [
    oec[oeci] = $ff;
    erase(oec);
  ]

  end;


erase(byte ec[])
  var eci, cb, i;

  begin

  eci = 0;
  while ((cb = ec[eci++]) != $ff) [
    if ((i = cubestat[cb][atr]) < $80) [
      if (i == 1)  mads--;
      cubestat[cb][atr] = $83;
      i = &map[cubestat[cb][posy]][cubestat[cb][posx]];
      memw[i] = memw[i+24] = $7a7a;
    ]
  ]
  end;


fallexpld(pcb)
  var i, j, a, x, y, cci;

  begin

  cci = 0;
  if (pcb != $ff)  chkcube[cci++] = pcb;

  ldir(map,mapcopy,24*24);

  for i=cubes-1 downto 0 [
    if (i != pcb) [
      case (a = cubestat[i][atr]) [
        $ff:;
        0 to $7f:[
          x = cubestat[i][posx];
          y = cubestat[i][posy];
          if (memw[&mapcopy[y+2][x]] == $7979) [
            memw[&map[y][x]] = $7979;
            memw[&map[y+1][x]] = $0080
            or (memw[&map[y+2][x]] = (i << 8) + i);
            cubestat[i][posy]++;
            chkcube[cci++] = i;
            locate(x+2,y);
            print("  ¥d¥l¥l",!(&cubeshp[a][0]));
          ]
        ]
        others:[
          x = cubestat[i][posx];
          y = cubestat[i][posy];
          j = a and $0f;
          locate(x+2,y);
          print(!(&explosion[j][0]));
          if (j) [
            cubestat[i][atr]--;
          ] else [
            cubestat[i][atr] = $ff;
            j = &map[y][x];
            memw[j] = memw[j+24] = $7979;
          ]
        ]
      ]
    ]
  ]
```

▶ 5-4.9-0.1。X680x0の電卓でやってみると面白いかもしれませんよ。X-BASICでやってもいいですけど。倒れてもいいように水枕を用意しておいたほうがいいかもしれませんね。X68030ではどうなるかわかりませんけど。　　　山崎　幹生(21)新潟県

```
  chkcube[cci] = $ff;

  end;


prtmads()

  begin

  if (mads != pmads) [
    locate(34,23);
    print(form$(pmads = mads,3));
    beep();
  ]

  end;


load()
  begin

  initscr();
  if (inputcbs() == false)  loadcbs();

  end;


edit()
  array byte m[3][8] = [
             "Rnd.Edit¥0",
             "Stg.Edit¥0",
             "Load    ¥0",
             "Save    ¥0"
          ];

  var i;

  begin

  initscr();

  loop [
    for i=0 to 3 [
      locate(29,i+7);
      print(!(m[i]));
    ]
    case (selectmenu(4)) [
      0: rndedit();
      1: stgedit();
      2: if (inputcbs() == false)  loadcbs();
      3: if (inputcbs() == false)  savecbs();
      others: exit;
    ]
    for i=1 to 22 [
      locate(3,i);
      print(spc$(22));
    ]
  ]

  end;


rndedit()
  array byte m[2][8] = [
             "Field   ¥0",
             "Cube    ¥0",
             "Round   ¥0"
          ];

  var i;

  begin

  initwork();
  makescr();
  edx = edy = edi = 1;
  for i=0 to 2 [
    locate(29,i+7);
    print(!(m[i]));
  ]
  locate(29,10);
  print(spc$(4));

  loop [
    case (selectmenu(3)) [
      0: landedit();
      1: cubeedit();
      2: chground();
      others: if (chkmc() == false)  exit;
    ]
  ]

  locate(34,18);
  print(spc$(3));

  end;


landedit()
  var dir, pdir, i;

  begin

  pdir = 1;

  loop [
    locate(edx+2,edy);
    case (i = inkey1()) [
      0 to 3: dir = pdir = i;
      4:[
        if (((i = getrnddt(edx,edy)) or getrnddt(edx-1,edy-1)
             or getrnddt(edx,edy-1) or getrnddt(edx-1,edy))
           <= 1) [
          putrnddt(edx,edy,i xor 1);
          print(str$(i ? ' ' : $7b,1));
          dir = pdir;
        ] else [
          beep();
          dir = 4;
        ]
      ]
      5: exit;
      others: dir = 4;
    ]
    case (dir) [
      0: if (--edx ==  0)  edx = 22;
      1: if (++edx == 23)  edx =  1;
      2: if (--edy ==  0)  edy = 22;
      3: if (++edy == 23)  edy =  1;
    ]
  ]

  end;


cubeedit()
  var i;

  begin

  locate(29,22);
  print("Cube:");
  prtcbmc(edi);

  loop [
    loop [
      locate(28,22);
      case (inkey1()) [
        0,2:[
          if (--edi == -1)  edi = 5;
          locate(34,22);
          prtcbmc(edi);
        ]
        1,3:[
          if (++edi == 6)  edi = 0;
          locate(34,22);
          prtcbmc(edi);
        ]
        4: exit;
        5: exit(2);
      ]
    ]

    loop [
      locate(edx+2,edy);
      case (i = inkey1()) [
        0: if (--edx ==  0)  edx = 22;
        1: if (++edx == 23)  edx =  1;
        2: if (--edy ==  0)  edy = 22;
        3: if (++edy == 23)  edy =  1;
        4:[
          if (getrnddt(edx-1,edy-1) or getrnddt(edx,edy-1)
              or getrnddt(edx+1,edy-1) or getrnddt(edx-1,edy)
              or getrnddt(edx-1,edy+1)) > 1 [
            beep();
          ]
          ef (getrnddt(edx+1,edy) or getrnddt(edx,edy+1)
            or getrnddt(edx+1,edy+1)) [
            beep();
          ] else [
            case (i = getrnddt(edx,edy)) [
              0:[
                putrnddt(edx,edy,edi+2);
                prtcbmc(edi);
              ]
              1: beep();
              others:[
                putrnddt(edx,edy,0);
                print(!(explosion));
                edi = i - 2;
                locate(34,22);
                prtcbmc(edi);
              ]
            ]
          ]
        ]
        5: exit;
      ]
    ]
  ]

  locate(29,22);
  print(spc$(7),"¥d¥l¥l  ");

  end;


prtcbmc(atr)

  begin

  if (atr) [
    print(!(&cubeshp[atr-1][0]));
  ] else [
    print(!(myshp));
  ]

  end;


getrnddt(x,y)
  var rd;

  begin

  if (x <  1)  return(1);
  if (x > 22)  return(1);
  if (y <  1)  return(1);
  if (y > 22)  return(1);
  rd = rnddt[y - 1][(x - 1) >> 1];

  end((x and 1) ? (rd >> 4) : (rd and $0f));


putrnddt(x,y,d)
  var byte rd[];
```


▶「階段落ち」を体験してしまいました。といっても記憶にないのですが，気を失ってそのままドドドッ！　となったようです。疲れているのでしょうか？　皆さん，体は大切にしましょう。
中矢　史朗(24)愛媛県

```
  begin

  rd = &rnddt[y - 1][(x - 1) >> 1];
  rd[0] = (x and 1) ? ((rd[0] and $0f) + (d << 4))
                    : ((rd[0] and $f0) + d);

  end;


chground()
  begin

  if (chkmc()) return;
  if (selectround() == false) [
    initwork();
    makescr();
    edx = edy = edi = 1;
  ]

  end;


chkmc()
  var x, y, rdt, cdt, me, cb, i;

  begin

  me = cb = i = 0;
  for y=1 to 22 [
    for x=1 to 22 [
      if (i = not i) [
        rdt = rnddt[y-1][x >> 1];
        cdt = rdt >> 4;
      ] else [
        cdt = rdt and $0f;
      ]
      case (cdt) [
        2     : me++;
        3 to 7: cb = true;
      ]
    ]
  ]
  if (me == 1)  if (cb)  return(false);
  beep();

  end(true);


stgedit()
  array byte m[3][8] = [
          "Move    ¥0",
          "Insert  ¥0",
          "Delete  ¥0",
          "Title   ¥0"
        ];

  var i;

  begin

  edx = edy = edi = edj = 0;
  for i=maxround-1 downto 0  eda[i] = i + 1;
  prtlist();

  for i=0 to 3 [
    locate(29,i+7);
    print(!(m[i]));
  ]

  loop [
    case (selectmenu(4)) [
      0: semove();
      1: seins();
      2: sedel();
      3: setitle();
      others: exit;
    ]
  ]

  rounddata[0] = maxround;

  end;


semove()
  array byte mbuf[]:cubestat;  /* for cutting down       */
                               /*   the size of this program */
  var rd, rn, i;

  begin

  edj = 0;

  loop [
    edi = 0;
    if ((rd = seselect()) == 0)  exit;
    ldir(rounddata + 17 + (rd - 1) * 242,mbuf,242);
    rn = eda[rd-1];
    rounddel(rd,1);
    edi = 1;
    if (i = seselect())  rd = i;
    roundins(rd);
    ldir(mbuf,rounddata + 17 + (rd - 1) * 242,242);
    eda[rd-1] = rn;
  ]

  end;


seins()
  var mm, rd;

  begin

  if ((mm = (_MEMAX - rounddata - 17) / 242) > 255)  mm = 255;
  edi = 1;
  edj = 0;

  while (rd = seselect()) [
    if (maxround >= mm) [
      locate(10,12);
      print(" Limit ¥1");
      beep();
      inkey(1);
    ] else [
      roundins(rd);
      newround(rd);
    ]
  ]

  end;


sedel()
  var rd, i;

  begin

  edi = 0;

  loop [
    edj = 0;
    if ((edj = seselect()) == 0)  exit;
    if (rd = seselect()) [
      if (rd < edj) [
        i = rd;
        rd = edj;
        edj = i;
      ]
      rounddel(rd,rd-edj+1);
      if (maxround == 0) [
        newround(1);
        maxround = 1;
      ]

      if (edj > maxround)  edj = maxround;
      i = edj - 1;
      edx = (i = edj - 1) / 20;
      edy = i - edx * 20;
    ]
  ]

  if (round > maxround)  round = maxround;

  end;


seselect()
  var maxx, lmaxy, i;

  begin

  prtlist();

  maxx = (i = maxround + edi - 1) / 20;
  lmaxy = i - maxx * 20;

  loop [
    locate(12,edy+2);
    case (inkey1()) [
      0:[
        if (--edx == -1) [
          edx = maxx;
          if (edy > lmaxy)  edy = lmaxy;
        ]
        prtlist();
      ]
      1:[
        if (++edx > maxx)  edx = 0;
        if (edx == maxx)  if (edy > lmaxy)  edy = lmaxy;
        prtlist();
      ]
      2:[
        if (--edy == -1) [
          edy = 19;
          if (--edx == -1) [
            edx = maxx;
            edy = lmaxy;
          ]
          prtlist();
        ]
      ]
      3:[
        if (++edy > ((edx == maxx) ? lmaxy : 19)) [
          edy = 0;
          if (++edx > maxx)  edx = 0;
          prtlist();
        ]
      ]
      4: exit;
      5: return(0);
    ]
  ]

  end(edx * 20 + edy + 1);


prtlist()
  var rd, i, j;

  begin

  rd = edx * 20 + 1;

  for i=0 to 19 [
    locate(9,i+2);
    if (rd <= maxround) [
      print(form$(rd,3),str$((rd == edj) ? '*' : ':',1),"[");
      if (j = eda[rd-1]) [
        print(form$(j,3));
      ] else [
        print("New");
      ]
      print("]");
    ]
    ef ((rd == maxround + 1) and edi) [
      print("---:-----");
    ] else [
```

▶裏をかかれるPictパズル……。ヤキイモと思えば落ち葉，トランシーバーと思えば携帯電話，お魚と思えば……タイヤキ(涙)。　中嶋　弘太郎(23)愛知県

```
        print(spc$(9));
      ]
      rd++;
    ]

    end;


roundins(rd)
  var rds, i;

  begin

  if (rds = ++maxround - rd) [
    i = rounddata + 17 + maxround * 242 - 1;
    lddr(i-242,i,rds*242);
    i = eda + maxround - 1;
    lddr(i-1,i,rds);
  ]

  end;


rounddel(rd,dels)
  var rds, i;

  begin

  if (rds = maxround - rd) [
    i = rounddata + 17 + rd * 242;
    ldir(i,i-dels*242,rds*242);
    i = eda + rd;
    ldir(i,i-dels,rds);
  ]
  maxround = maxround - dels;

  end;


newround(rd)

  begin

  rnddt = rounddata + 17 + (rd - 1) * 242;
  rnddt[0][0] = 0;
  ldir(rnddt,rnddt+1,241);
  rnddt[20][ 9] = $40;
  rnddt[20][10] = $20;
  eda[rd-1] = 0;

  end;


setitle()
  array byte savedtitle[15];

  var cx, i;

  begin

  for i=0 to 15  savedtitle[i] = rounddata[i+1];
  cx = 0;

  loop [
    locate((cx and 7) + 28,(cx >> 3) + 13);
    case (i = inkey(1)) [
      $20 to $ff:[
        rounddata[cx+1] = i;
        if (++cx == 16)  cx--;
      ]
      '¥l': if (--cx == -1)  cx++;
      '¥r': if (++cx == 16)  cx--;
      '¥u':[
        if (--cx == -1)  cx++;
        for i=cx+1 to 15  rounddata[i] = rounddata[i+1];
        rounddata[16] = ' ';
      ]
      '¥d':[
        if (cx < 15) [
          for i=15 downto cx+1  rounddata[i+1] = rounddata[i];
        ]
        rounddata[cx+1] = ' ';
      ]
      '¥c':[
        for i=1 to 16  rounddata[i] = ' ';
        cx = 0;
      ]
      '¥n': return;
      $1b: exit;
    ]
    prtstgtitle();
  ]

  for i=0 to 15  rounddata[i+1] = savedtitle[i];
  prtstgtitle();

  end;


selectmenu(items)
  var i;

  begin

  for i=1 to items [
    locate(28,i+6);
    print(" ");
  ]
  i = 0;
  loop [
    locate(28,i+7);
    case (inkey1()) [
      2: if (--i == -1)  i = items - 1;
      3: if (++i ==  items)  i = 0;
      4: exit;
      5: return($ff);
    ]
  ]
  print("*");


  end(i);


initscr()
  var i;

  begin

  print("¥c");

  locate(2,0);
  print(str$($7b,24));
  for i=1 to 22 [
    locate(2,i);
    print(str$($7b,1),tab$(22),str$($7b,1));
  ]
  locate(2,23);
  print(str$($7b,24));

  locate(28,0);
  print(str$('ロ',9));
  locate(29,2);
  print("C U B E");
  locate(28,4);
  print(str$('ロ',9));
  locate(28,18);
  print("Round");
  prtstgtitle();

  end;


selectround()
  var r, i, j;

  begin

  locate(34,18);
  print(form$(r = round,3));

  loop [
    locate(27,18);
    case (inkey1()) [
      0,2:[
        if (--r < 1)  r = maxround;
        locate(34,18);
        print(form$(r,3));
      ]
      1,3:[
        if (++r > maxround)  r = 1;
        locate(34,18);
        print(form$(r,3));
      ]
      4: exit;
      5:[
        locate(34,18);
        print(form$(round,3));
        return(true);
      ]
    ]
  ]
  round = r;

  end(false);


initwork()
  var cb, rdt, cdt, x, y, i;

  begin

  for i=0 to 23 [
    map[0][i] = map[23][i] = $7a;
    map[1][i] = $79;
  ]
  map[1][0] = map[1][23] = $7a;
  ldir(&map[1][0],&map[2][0],24*21);

  rnddt = rounddata + 17 + (round - 1) * 242;
  mads = cb = i = 0;
  for y=1 to 22 [
    for x=1 to 22 [
      if (i = not i) [
        rdt = rnddt[y-1][x >> 1];
        cdt = rdt >> 4;
      ] else [
        cdt = rdt and $0f;
      ]
      case (cdt) [
        1: map[y][x] = $7a;
        2:[
          myposx = x;
          myposy = y;
          memw[&map[y][x]] = memw[&map[y+1][x]] = $7b7b;
        ]
        3 to 7:[
          cubestat[cb][posx] = x;
          cubestat[cb][posy] = y;
          cubestat[cb][atr] = cdt - 3;
          memw[&map[y][x]] = $0080
          or (memw[&map[y+1][x]] = (cb << 8) + cb);
          cb++;
          if (cdt == 4)  mads++;
        ]
      ]
    ]
  ]
  cubes = cb;
  pmads = mads;

  end;


makescr()
  var i, j;

  begin

```

▶はうっ，ついにメモリが2Mバイトに!!　あまりの広大さにクラクラです。「12Mバイトでも足りん」というが多いですけど，私にはこれで十分です。PCM 8 とASKが同時に常駐できるなんて，ライクアドリームだわ(錯乱中)。　大谷　一友(20)兵庫県

```
  for i=1 to 22 [
    for j=1 to 22[
      locate(j+2,i);
      print(str$((map[i][j] == $7a) ? $7b : ' ',1));
    ]
  ]

  locate(myposx+2,myposy);
  print(!(myshp));

  for i=cubes-1 downto 0 [
    locate(cubestat[i][posx]+2,cubestat[i][posy]);
    print(!(&cubeshp[cubestat[i][atr]][0]));
  ]

  locate(34,18);
  print(form$(round,3));

  end;


inputcbs()
  var cx, i;

  begin

  locate(4,11);
  print("File Name:");
  locate(5,12);
  print("[",!(inputbuf),".CBS]");

  cx = 0;
  loop [
    locate(cx+6,12);
    case (i = inkey(1)) [
      ':','.':;
      $20 to $ff:[
        inputbuf[cx] = i;
        if (++cx == 13)  cx--;
      ]
      '¥l': if (--cx == -1)  cx++;
      '¥r': if (++cx == 13)  cx--;
      '¥u':[
        if (--cx == -1)  cx++;
        for i=cx to 11  inputbuf[i] = inputbuf[i+1];
        inputbuf[12] = ' ';
      ]
      '¥d':[
        if (cx < 12) [
          for i=12 downto cx+1  inputbuf[i] = inputbuf[i-1];
        ]
        inputbuf[cx] = ' ';
      ]
      '¥c':[
        for i=0 to 12  inputbuf[i] = ' ';
        cx = 0;
      ]
      '¥n': exit;
      $1b: return(true);
    ]
    locate(6,12);
    print(!(inputbuf));
  ]

  for i=0 to 12  accesscbs[i] = inputbuf[i];

  end(false);


loadcbs()
  var i;

  begin

  ^A  = 1;
  ^DE = accesscbs;
  call(_FILE);
  repeat [
    call(_ROPEN);
    if (^CY == 1) [
      locate(7,14);
      print("File Not Found ¥l");
      beep();
      inkey(1);
      return(true);
    ]
  ] until (^ZERO == 1);

  if (_SIZE > _MEMAX - rounddata) [
    locate(7,14);
    print("Too Long File ¥l");
    beep();
    inkey(1);
    return(true);
  ]

  _DTADR = rounddata;
  call(_RDD);
  if (^CY == 1) [
    locate(6,14);
    print("File Read Error ¥l");
    beep();
    inkey(1);
    return(true);
  ]

  round = 1;
  maxround = rounddata[0];
  prtstgtitle();

  end(false);


savecbs()
  begin

  ^A  = 1;
  ^DE = accesscbs;
  call(_FILE);

  _SIZE = 17 + maxround * 242;
  _DTADR = rounddata;
  _EXADR = 0;

  call(_WOPEN);
  if (^CY == 0)  call(_WRD);
  if (^CY == 1) [
    locate(6,14);
    print("File Write Error ¥l");
    beep();
    inkey(1);
  ]

  end;


prtstgtitle()
  var i;

  begin

  locate(28,13);
  for i=1 to 8  print(str$(rounddata[i],1));
  locate(28,14);
  for i=9 to 16  print(str$(rounddata[i],1));

  end;


inkey1()
  var ky, i;

  begin

  while (inkey(0) == $1b) []

  ky = inkey(1);
  for i=0 to 23 [
    if (ky == keyassign[i])  exit;
  ]

  end(i >> 2);


ldir(3)
  begin

  code($ed,$b0);  /* LDIR */

  end;


lddr(3)
  begin

  code($ed,$b8);  /* LDDR */

  end;


wait()
  var i;

  begin

  for i=0 to waitcnt []

  end;
```

リスト2 CUBE.CBS

```
7000 10 4D 65 72 63 75 72 79 : F7
7008 20 20 20 20 53 74 61 67 : 0F
7010 65 00 00 00 00 00 00 00 : 65
7018 00 00 00 00 00 00 00 00 : 00
7020 00 00 00 00 00 00 00 00 : 00
7028 06 04 00 60 00 00 00 00 : 6A
7030 00 00 00 00 00 00 00 00 : 00
7038 00 00 00 00 00 00 11 11 : 22
7040 11 11 00 00 00 00 00 00 : 22
7048 00 00 00 00 00 00 00 00 : 00
7050 00 00 00 00 00 00 00 00 : 00
7058 00 00 70 00 00 00 00 00 : 70
7060 00 04 00 00 00 00 00 00 : 04

7068 00 00 00 00 00 00 00 11 : 11
7070 11 11 11 00 00 00 00 11 : 44
7078 11 00 11 11 11 11 00 00 : 55
---------------------------------
SUM: CE 97 17 03 C7 FA E4 13 AFA3

7080 00 00 11 11 00 11 11 11 : 55
7088 11 00 00 00 00 11 11 00 : 33
7090 11 11 11 11 00 00 00 00 : 44
7098 11 11 00 11 11 11 11 00 : 66
70A0 00 00 00 00 00 70 00 00 : 70
70A8 00 00 00 50 00 00 00 00 : 50
70B0 00 00 00 00 00 00 00 00 : 00

70B8 00 11 11 11 11 11 11 00 : 66
70C0 00 04 00 00 11 11 11 11 : 48
70C8 11 11 00 00 00 00 00 00 : 22
70D0 00 00 00 00 00 00 00 04 : 04
70D8 00 00 00 00 00 00 00 00 : 00
70E0 00 00 00 00 00 00 00 00 : 00
70E8 00 00 00 00 00 50 00 00 : 50
70F0 00 00 20 00 00 00 00 00 : 20
70F8 00 00 00 00 00 00 00 00 : 00
---------------------------------
SUM: 44 48 53 94 33 15 55 26 F270

7100 00 00 00 20 00 00 00 00 : 20
```

▶秋葉原に行った(交通費，約25,000円)。RGB切り換え機(1,200円)を買って帰ってきた。負けへんで！　平　勝久(21)大阪府

```
7108 00 00 00 00 00 00 00 00 : 00
7110 00 00 00 00 30 00 06 00 : 36
7118 00 00 00 00 00 00 00 00 : 00
7120 00 00 00 00 00 00 00 00 : 00
7128 00 00 11 00 11 10 00 00 : 32
7130 00 00 00 00 00 11 00 11 : 22
7138 10 00 00 00 00 00 00 00 : 10
7140 11 11 11 10 00 00 00 40 : 83
7148 00 00 00 11 11 11 10 00 : 43
7150 00 00 00 00 00 00 00 00 : 00
7158 00 00 00 00 00 11 00 00 : 11
7160 00 00 00 00 00 00 00 00 : 00
7168 11 00 00 00 00 00 00 00 : 11
7170 00 00 00 11 11 11 11 11 : 55
7178 11 11 00 00 00 00 11 11 : 44
---------------------------------
SUM: 43 22 22 52 63 54 38 73 07D9

7180 11 11 11 11 11 00 00 00 : 55
7188 00 00 00 00 00 00 00 00 : 00
7190 00 00 00 40 00 00 00 00 : 40
7198 00 70 00 00 00 00 00 00 : 70
71A0 00 00 00 00 00 00 00 00 : 00
71A8 11 11 10 01 11 00 11 11 : 66
71B0 11 10 00 11 11 10 01 11 : 65
71B8 00 11 11 11 10 00 11 11 : 65
71C0 10 01 11 11 11 11 11 10 : 76
71C8 00 11 11 10 01 11 11 11 : 66
71D0 11 11 10 00 11 11 10 00 : 64
71D8 00 00 00 00 00 00 00 11 : 11
71E0 11 10 00 00 06 00 00 00 : 27
71E8 00 00 11 11 10 00 00 00 : 32
71F0 00 00 00 00 00 00 00 00 : 00
71F8 00 00 00 00 00 00 00 00 : 00
---------------------------------
SUM: 65 E6 75 A6 7C 43 55 65 4AA5

7200 00 00 00 00 00 00 00 00 : 00
7208 00 00 00 00 00 00 00 00 : 00
7210 00 11 11 11 11 11 00 00 : 55
7218 00 00 00 00 00 40 00 00 : 40
7220 00 00 00 00 00 00 00 00 : 00
7228 00 00 00 00 00 00 00 00 : 00
7230 00 00 00 50 00 00 00 00 : 50
7238 40 00 00 00 00 00 00 00 : 40
7240 00 00 00 00 00 00 00 00 : 00
7248 11 11 11 11 11 00 30 00 : 85
7250 00 10 00 00 00 00 00 00 : 10
7258 00 00 00 00 10 00 00 00 : 10
7260 00 00 00 11 11 11 11 10 : 54
7268 00 00 00 00 00 00 00 00 : 00
7270 00 00 00 00 00 20 00 00 : 20
7278 00 00 00 00 00 00 00 00 : 00
---------------------------------
SUM: 51 32 22 83 43 82 41 10 D6DC

7280 00 00 00 00 00 00 00 00 : 00
7288 00 00 00 00 00 00 00 00 : 00
7290 00 00 00 00 00 00 00 00 : 00
7298 00 00 00 00 11 11 11 11 : 44
72A0 11 11 11 11 11 00 00 00 : 55
72A8 00 00 00 00 00 00 00 00 : 00
72B0 00 00 00 00 00 00 00 00 : 00
72B8 00 00 00 00 00 00 00 00 : 00
72C0 06 00 00 00 00 00 00 00 : 06
72C8 00 00 00 00 00 00 00 00 : 00
72D0 00 00 00 00 00 00 00 70 : 70
72D8 00 00 40 70 00 00 00 00 : B0
72E0 00 00 00 00 00 00 00 00 : 00
72E8 00 00 00 00 00 00 00 00 : 00
72F0 00 20 00 00 00 00 00 00 : 20
72F8 00 00 00 00 00 00 00 00 : 00
---------------------------------
SUM: 17 31 51 81 22 11 11 81 5F2E

7300 00 00 00 00 00 00 00 00 : 00
7308 00 50 00 00 07 00 00 00 : 57
7310 00 00 00 00 00 00 00 00 : 00
7318 00 00 00 00 00 00 00 11 : 11
7320 11 11 11 11 11 10 00 00 : 65
7328 00 00 00 00 00 00 00 30 : 30
7330 00 00 00 00 00 00 00 00 : 00
7338 00 00 00 00 00 00 00 50 : 50
7340 40 00 11 11 11 30 11 11 : C5
7348 11 00 00 00 00 01 11 11 : 34
7350 00 00 00 00 00 11 11 00 : 22
7358 40 50 11 30 00 00 00 00 : D1
7360 11 10 00 00 00 11 00 00 : 32
7368 00 00 00 50 40 00 11 11 : B2
7370 11 30 00 00 00 00 00 00 : 41
7378 00 01 11 11 00 00 00 00 : 23
---------------------------------
SUM: C4 F2 44 B3 69 63 44 C4 1557

7380 00 11 11 00 40 50 11 03 : C6
7388 00 00 00 00 11 10 00 00 : 21
7390 00 11 00 00 00 00 00 50 : 61
7398 40 00 11 11 11 03 00 00 : 76
73A0 00 00 00 00 00 01 11 11 : 23
73A8 00 00 00 00 00 11 11 00 : 22
73B0 40 50 11 03 00 00 00 00 : A4
73B8 11 10 00 00 00 11 00 00 : 32
73C0 00 00 00 50 40 00 11 11 : B2
73C8 16 03 00 06 00 00 00 00 : 1F
73D0 00 01 11 10 00 00 00 00 : 22
73D8 00 20 00 00 00 00 03 00 : 23
73E0 00 00 00 00 00 00 00 00 : 00
73E8 00 00 00 00 00 00 00 00 : 00
73F0 00 00 00 00 07 00 00 00 : 07
73F8 00 00 00 00 00 00 00 00 : 00
---------------------------------
SUM: A7 A6 44 7A A9 86 47 75 FDBA

7400 00 00 00 00 00 00 00 00 : 00
7408 00 00 01 11 11 11 11 11 : 56
7410 00 00 00 00 00 01 11 11 : 23
7418 11 11 11 00 00 00 00 01 : 34
7420 11 11 11 11 11 11 00 00 : 66
7428 00 00 01 11 11 11 11 11 : 56
7430 11 00 00 00 00 00 00 00 : 11
7438 00 00 00 00 00 00 00 00 : 00
7440 00 00 00 00 00 00 00 00 : 00
7448 00 50 00 00 00 00 00 00 : 50
7450 50 00 00 00 00 00 00 00 : 50
7458 00 00 00 00 00 11 11 11 : 33
7460 11 00 00 00 00 00 40 00 : 51
7468 11 11 11 11 00 00 00 00 : 44
7470 00 00 00 00 00 00 00 00 : 00
7478 00 11 00 00 11 00 00 00 : 22
---------------------------------
SUM: A5 94 35 44 44 45 84 45 29AA

7480 00 00 00 00 11 00 00 11 : 22
7488 00 00 00 00 00 00 00 11 : 11
7490 11 11 11 00 00 00 00 00 : 33
7498 00 00 11 11 11 11 00 00 : 44
74A0 70 00 00 00 00 00 00 00 : 70
74A8 00 00 00 00 00 00 00 00 : 00
74B0 00 00 00 00 00 00 40 40 : 80
74B8 00 00 00 00 40 04 07 00 : 4B
74C0 00 00 00 00 00 00 00 00 : 00
74C8 00 00 00 00 06 00 00 00 : 06
74D0 60 00 11 11 11 11 00 00 : A4
74D8 00 00 00 00 00 00 01 11 : 12
74E0 11 00 11 10 00 00 50 00 : 82
74E8 00 00 00 11 00 11 11 00 : 33
74F0 00 00 00 00 00 00 01 00 : 01
74F8 01 11 00 00 11 00 07 00 : 2A
---------------------------------
SUM: F3 22 44 43 8A 37 B1 73 CB4B

7500 00 00 00 00 00 00 00 11 : 11
7508 10 00 00 00 00 00 00 00 : 10
7510 00 01 10 00 00 60 00 00 : 71
7518 00 40 00 00 01 00 00 00 : 41
7520 00 00 00 00 00 00 00 00 : 00
7528 00 01 11 11 11 30 00 70 : D4
7530 00 00 00 00 00 00 01 11 : 12
7538 00 00 00 00 00 00 00 00 : 00
7540 00 00 01 70 00 11 11 00 : 93
7548 11 11 10 00 00 00 00 00 : 32
7550 11 11 30 11 10 00 00 00 : 73
7558 00 01 00 01 11 00 11 00 : 24
7560 00 02 00 00 01 00 01 11 : 15
7568 30 11 00 00 00 00 00 00 : 41
7570 00 00 11 00 10 00 00 00 : 21
7578 00 00 00 00 00 01 30 10 : 41
---------------------------------
SUM: 62 78 73 93 44 A2 54 B3 DBDD

7580 00 00 00 00 00 00 00 00 : 00
7588 00 00 00 00 04 00 00 00 : 04
7590 00 00 00 00 30 00 00 00 : 30
7598 00 00 00 00 00 00 00 00 : 00
75A0 00 00 50 00 00 60 00 10 : C0
75A8 00 05 00 00 00 00 00 00 : 05
75B0 00 00 11 00 00 01 11 11 : 34
75B8 11 00 00 01 11 20 00 00 : 43
75C0 07 00 00 30 00 40 00 00 : 77
75C8 00 00 00 00 00 00 00 00 : 00
75D0 00 00 00 00 00 00 01 10 : 11
75D8 01 11 00 11 11 00 00 00 : 34
75E0 00 01 10 01 11 00 11 11 : 45
75E8 00 00 00 00 01 11 11 10 : 33
75F0 00 50 00 00 00 00 00 01 : 51
75F8 11 11 00 00 00 00 00 00 : 22
---------------------------------
SUM: 2A 78 71 43 68 D2 34 53 1482

7600 00 00 00 00 00 00 00 30 : 30
7608 00 00 00 50 00 00 00 00 : 50
7610 00 00 00 00 00 00 00 00 : 00
7618 00 00 00 00 00 11 00 00 : 11
7620 11 11 10 00 04 07 00 00 : 3D
7628 00 00 00 11 11 10 00 00 : 32
7630 00 00 00 00 00 00 00 00 : 00
7638 00 01 11 11 11 00 00 11 : 45
7640 11 00 00 00 01 11 11 11 : 45
7648 00 00 11 11 00 00 00 00 : 22
7650 00 00 00 00 00 00 00 00 : 00
7658 00 00 00 00 00 00 00 00 : 00
7660 00 00 00 00 11 11 11 10 : 43
7668 00 00 00 00 00 00 00 11 : 11
7670 11 11 11 11 00 00 00 00 : 44
7678 00 00 11 11 11 11 11 00 : 55
---------------------------------
SUM: 33 23 54 A5 49 5B 33 73 8AF3

7680 00 00 00 00 00 11 11 11 : 33
7688 11 11 11 00 00 00 00 00 : 33
7690 11 11 11 11 11 11 00 00 : 66
7698 00 00 00 00 00 00 00 00 : 00
76A0 03 03 05 00 00 00 00 00 : 0B
76A8 00 00 00 00 00 00 00 00 : 00
76B0 00 00 00 00 70 00 00 00 : 70
76B8 00 00 00 00 00 00 00 00 : 00
76C0 00 00 00 00 00 00 00 00 : 00
76C8 00 00 50 00 00 00 03 00 : 53
76D0 00 30 00 00 00 00 00 00 : 30
76D8 00 00 00 00 00 00 00 00 : 00
76E0 11 11 11 11 11 00 00 70 : C5
76E8 00 00 00 00 00 00 00 00 : 00
76F0 00 00 00 00 00 00 00 00 : 00
76F8 00 00 00 00 11 11 11 11 : 44
---------------------------------
SUM: 36 66 88 22 A3 33 25 92 FFCF

7700 00 00 50 00 00 00 00 11 : 61
7708 11 11 11 00 00 00 00 00 : 33
7710 00 00 00 00 00 11 00 11 : 22
7718 11 11 00 00 00 50 04 00 : 76
7720 11 00 11 11 11 00 00 00 : 44
7728 00 00 00 11 11 11 11 11 : 55
7730 11 00 00 70 30 30 30 30 : 41
7738 30 30 30 30 00 00 00 00 : C0
7740 00 00 00 00 00 00 00 00 : 00
7748 00 70 10 01 11 11 10 06 : B9
7750 00 01 00 00 00 00 00 07 : 08
7758 00 00 00 00 01 00 00 60 : 61
7760 00 00 00 00 00 01 10 01 : 12
7768 00 00 00 00 01 11 11 10 : 33
7770 00 00 00 00 00 30 00 00 : 30
7778 00 00 00 00 00 00 00 00 : 00
---------------------------------
SUM: 74 C3 B2 C3 65 F5 76 E1 BA57

7780 00 00 00 00 00 00 00 00 : 00
7788 00 00 00 06 00 01 11 10 : 28
7790 05 00 00 00 00 20 00 00 : 25
7798 01 11 10 00 00 00 00 00 : 22
77A0 00 20 00 06 00 00 00 00 : 26
77A8 30 40 30 00 00 00 00 00 : A0
77B0 00 00 00 00 00 00 00 00 : 00
77B8 00 05 00 00 00 03 00 00 : 08
77C0 05 00 00 00 00 00 00 00 : 05
77C8 00 00 00 00 00 00 00 11 : 11
77D0 11 11 11 00 01 10 01 11 : 56
77D8 00 00 00 70 00 07 00 00 : 77
77E0 60 00 00 00 00 00 00 00 : 60
77E8 00 00 00 00 00 00 00 00 : 00
77F0 00 06 04 03 03 00 30 00 : 40
77F8 00 00 00 00 00 00 00 00 : 00
---------------------------------
SUM: AC 8D 55 7F 04 38 45 32 B73D

7800 00 00 00 00 00 03 00 00 : 03
7808 30 30 30 00 10 00 00 00 : A0
7810 00 00 00 00 00 00 00 00 : 00
7818 00 00 00 00 11 10 01 11 : 33
7820 11 00 00 00 00 00 00 00 : 11
7828 00 00 00 00 00 00 00 00 : 00
7830 00 00 00 00 00 00 00 00 : 00
7838 00 00 00 00 40 00 00 00 : 40
7840 00 11 00 00 00 00 00 00 : 11
7848 00 00 00 01 11 00 00 00 : 12
7850 00 00 07 00 00 00 11 11 : 29
7858 03 00 00 00 00 00 00 00 : 03
7860 00 11 10 00 00 00 00 00 : 21
7868 04 00 00 00 60 40 04 00 : A8
7870 00 00 00 00 00 00 00 00 : 00
7878 00 00 00 00 00 00 01 11 : 12
---------------------------------
SUM: 48 52 47 01 D2 53 17 33 7E45

7880 00 00 11 10 05 00 05 00 : 2B
7888 00 01 11 10 01 11 10 00 : 44
7890 00 00 00 00 00 00 00 00 : 00
7898 01 00 00 00 00 00 00 00 : 01
78A0 00 00 00 01 00 00 00 00 : 01
78A8 00 00 00 00 00 00 01 00 : 01
78B0 00 00 30 00 00 06 00 00 : 36
78B8 00 01 00 00 00 00 00 00 : 01
78C0 00 00 00 00 01 00 00 00 : 01
78C8 30 00 00 04 00 00 00 01 : 35
78D0 00 00 00 00 00 00 00 00 : 00
78D8 00 00 01 00 00 00 30 00 : 31
78E0 00 07 00 00 00 01 00 00 : 08
78E8 00 00 00 00 00 00 00 00 : 00
78F0 01 00 00 00 30 00 00 70 : A1
78F8 50 05 00 01 00 00 00 00 : 56
---------------------------------
SUM: 82 0E 53 26 37 18 46 71 2BE2

7900 00 00 00 00 00 00 01 00 : 01
7908 00 00 01 00 00 10 01 11 : 23
7910 11 11 00 00 01 11 00 00 : 34
7918 00 00 00 00 00 00 00 00 : 00
7920 00 00 00 00 00 00 00 00 : 00
7928 00 00 70 00 00 00 00 00 : 70
7930 00 00 20 00 00 00 00 00 : 20
7938 00 00 00 00 00 00 00 00 : 00
7940 60 00 60 00 00 00 00 00 : C0
```

▶せっかく不安を煽ってもらったのにすいませんが，僕や友人たちはすべて内定をもらえました。あ，いや，いいんですよ，お祝いなんて。ましてMach-2がほしいなんてひと言も……。

今井　健生(23)奈良県

```
7948 00 00 00 00 00 00 11 11 : 22
7950 11 11 10 00 00 00 11 00 : 43
7958 11 00 00 00 04 00 00 00 : 15
7960 01 11 40 11 00 00 00 00 : 63
7968 00 00 00 11 11 00 11 07 : 3A
7970 07 00 04 00 00 01 11 11 : 2E
7978 70 11 00 00 00 00 00 00 : 81
---------------------------------
SUM: 0B 44 45 22 16 22 46 3A 9783

7980 11 11 11 00 11 00 70 00 : B4
7988 00 00 03 00 00 05 00 00 : 08
7990 00 00 00 00 00 00 00 00 : 00
7998 00 00 00 00 11 11 10 00 : 32
79A0 11 11 00 11 11 00 00 11 : 55
79A8 11 10 00 11 11 00 11 11 : 65
79B0 00 00 15 00 10 00 11 11 : 47
79B8 11 11 11 00 00 00 00 00 : 33
79C0 00 11 11 13 01 11 00 00 : 47
79C8 30 30 70 00 00 00 00 00 : D0
79D0 00 00 00 00 00 00 00 00 : 00
79D8 00 00 40 00 00 11 10 01 : 62
79E0 10 00 60 00 00 00 00 00 : 70
79E8 11 10 01 10 00 00 00 10 : 42
79F0 01 11 00 00 00 01 10 00 : 23
79F8 11 00 00 00 00 00 00 00 : 11
---------------------------------
SUM: A7 A5 5C 45 55 39 C2 44 F1E1

7A00 01 11 00 11 00 00 00 00 : 23
7A08 00 00 00 01 10 00 11 00 : 22
7A10 00 00 00 00 00 00 01 10 : 11
7A18 00 00 00 00 00 00 00 00 : 00
7A20 00 00 00 00 00 00 00 01 : 01
7A28 11 00 11 10 00 60 00 10 : A2
7A30 00 00 00 00 00 11 10 00 : 21
7A38 00 11 10 00 00 00 00 00 : 21
7A40 11 10 00 11 11 10 01 00 : 54
7A48 00 00 00 00 00 00 11 11 : 22
7A50 10 01 10 00 00 00 00 00 : 21
7A58 00 11 11 10 01 11 00 00 : 44
7A60 00 00 03 00 00 00 00 00 : 03
7A68 00 00 00 20 00 00 00 00 : 20
7A70 00 00 00 00 00 00 00 00 : 00
7A78 00 50 00 00 00 10 50 00 : B0
---------------------------------
SUM: 33 94 45 63 22 A2 84 32 C7EC

7A80 30 11 00 00 00 00 00 00 : 41
7A88 10 00 00 00 11 00 00 50 : 71
7A90 00 00 00 10 01 00 00 11 : 22
7A98 00 00 00 00 00 00 10 01 : 11
7AA0 60 00 70 00 00 01 11 11 : F3
7AA8 11 10 01 00 00 00 00 00 : 22
7AB0 00 06 00 00 04 01 00 00 : 0B
7AB8 11 00 00 00 00 00 00 00 : 11
7AC0 01 00 00 11 00 00 00 01 : 13
7AC8 10 00 14 01 00 00 11 00 : 36
7AD0 00 00 01 10 00 10 01 00 : 22
7AD8 00 50 00 40 00 00 00 00 : 90
7AE0 15 01 00 00 00 00 00 00 : 16
7AE8 00 00 00 10 01 00 00 11 : 22
7AF0 00 10 00 00 00 00 11 11 : 32
7AF8 00 00 11 00 10 00 11 11 : 43
---------------------------------
SUM: E8 88 97 82 27 12 55 A7 088A

7B00 00 01 11 00 00 11 11 10 : 44
7B08 01 11 11 00 00 11 00 00 : 34
7B10 60 11 00 00 00 00 00 00 : 71
7B18 11 00 00 00 00 00 00 00 : 11
7B20 00 00 00 11 00 00 11 00 : 22
7B28 00 40 00 60 30 00 11 00 : E1
7B30 00 11 00 00 00 00 00 00 : 11
7B38 00 11 00 00 11 11 00 11 : 44
7B40 11 11 11 11 11 00 00 00 : 55
```

```
7B48 11 00 11 11 11 11 11 11 : 77
7B50 00 00 00 00 00 00 00 00 : 00
7B58 00 00 30 20 00 00 00 00 : 50
7B60 00 00 00 00 00 00 00 00 : 00
7B68 00 00 50 00 00 00 00 00 : 50
7B70 00 00 00 00 00 00 00 00 : 00
7B78 00 00 00 00 00 00 00 00 : 00
---------------------------------
SUM: 94 96 C4 B3 63 44 44 32 82CB

7B80 50 00 00 00 20 00 00 00 : 70
7B88 40 00 00 00 00 00 00 00 : 40
7B90 00 00 00 00 00 00 11 00 : 11
7B98 00 00 00 00 00 00 11 11 : 22
7BA0 00 00 00 00 00 00 00 00 : 00
7BA8 00 00 00 00 00 00 00 00 : 00
7BB0 00 00 00 00 00 00 00 00 : 00
7BB8 40 00 03 00 00 00 01 11 : 55
7BC0 00 00 00 00 00 00 00 00 : 00
7BC8 00 11 01 00 00 00 11 11 : 34
7BD0 11 11 11 11 16 01 00 00 : 5B
7BD8 06 00 40 00 00 00 00 00 : 46
7BE0 01 00 00 00 00 00 50 00 : 51
7BE8 00 00 00 30 00 00 00 11 : 41
7BF0 11 00 11 11 11 10 00 00 : 54
7BF8 00 00 30 00 50 00 00 00 : 80
---------------------------------
SUM: F9 22 96 52 97 11 84 44 9C2E

7C00 00 30 00 00 00 00 00 00 : 30
7C08 00 00 00 00 00 00 00 00 : 00
7C10 11 11 30 11 10 00 00 60 : D3
7C18 00 00 07 00 00 00 00 11 : 18
7C20 00 00 00 00 00 10 00 00 : 10
7C28 60 00 01 11 11 10 00 00 : 93
7C30 60 00 00 00 00 00 00 50 : B0
7C38 00 00 00 00 00 00 30 00 : 30
7C40 00 00 00 00 00 00 70 00 : 70
7C48 00 00 00 00 00 50 40 00 : 90
7C50 00 00 00 00 11 00 00 00 : 11
7C58 00 00 00 20 00 00 00 07 : 27
7C60 00 00 04 00 00 00 00 00 : 04
7C68 00 00 00 00 00 00 00 00 : 00
7C70 00 11 11 00 00 01 11 00 : 34
7C78 11 10 00 00 00 30 00 00 : 51
---------------------------------
SUM: E2 62 4D 42 32 A1 F1 C8 8F83

7C80 06 00 00 00 00 00 00 00 : 06
7C88 00 00 00 00 00 00 00 00 : 00
7C90 00 00 00 11 00 00 01 11 : 23
7C98 00 00 00 00 00 00 40 00 : 40
7CA0 00 00 00 00 00 00 00 00 : 00
7CA8 00 00 00 00 00 00 00 60 : 60
7CB0 30 40 00 00 11 00 00 01 : 82
7CB8 11 00 00 00 00 00 00 30 : 41
7CC0 00 00 00 00 00 50 50 50 : F0
7CC8 60 00 00 00 00 00 00 00 : 60
7CD0 00 00 00 00 11 11 00 00 : 22
7CD8 01 11 00 30 70 70 50 00 : 72
7CE0 00 00 00 00 00 00 00 00 : 00
7CE8 00 00 00 00 00 00 00 00 : 00
7CF0 00 01 10 06 00 00 00 00 : 17
7CF8 00 00 00 00 00 00 00 00 : 00
---------------------------------
SUM: A8 52 10 47 92 D1 E1 F2 97C6

7D00 00 00 00 00 00 00 00 00 : 00
7D08 00 00 11 00 30 00 00 00 : 41
7D10 00 00 00 00 00 11 00 00 : 11
7D18 00 00 00 00 00 00 00 00 : 00
7D20 00 50 30 30 00 50 30 00 : 30
7D28 11 11 00 00 00 00 00 00 : 22
7D30 00 00 00 11 11 00 00 11 : 33
7D38 11 11 11 11 11 11 11 11 : 88
7D40 00 00 11 11 11 11 11 11 : 66
```

```
7D48 11 11 11 00 00 03 00 00 : 36
7D50 70 00 00 00 00 00 00 00 : 70
7D58 00 00 00 00 00 00 00 00 : 00
7D60 30 00 00 00 60 03 03 00 : 96
7D68 60 00 00 00 00 00 00 00 : 60
7D70 00 00 00 00 00 00 03 00 : 03
7D78 00 00 01 11 00 11 10 00 : 33
---------------------------------
SUM: 33 83 75 74 C3 9A 68 33 75ED

7D80 00 00 00 00 05 00 00 00 : 05
7D88 00 00 00 01 11 11 00 00 : 23
7D90 00 00 00 00 00 00 00 00 : 00
7D98 07 00 00 11 00 00 00 00 : 18
7DA0 00 00 00 00 00 00 11 00 : 11
7DA8 00 00 00 00 01 11 11 00 : 23
7DB0 00 00 00 00 00 00 00 00 : 00
7DB8 00 00 00 00 00 00 00 00 : 00
7DC0 00 00 00 00 00 00 00 00 : 00
7DC8 00 00 01 11 10 01 11 00 : 34
7DD0 00 00 00 00 00 01 11 10 : 22
7DD8 01 11 00 00 00 50 00 00 : 62
7DE0 00 00 00 00 11 00 00 00 : 11
7DE8 00 00 00 00 00 00 00 00 : 00
7DF0 00 00 11 11 16 05 00 00 : 3D
7DF8 00 00 00 00 00 11 11 10 : 32
---------------------------------
SUM: 08 11 12 34 4E 8A 55 20 6BBD

7E00 00 00 00 00 00 11 00 00 : 11
7E08 11 11 11 11 11 11 11 11 : 88
7E10 11 00 00 00 00 00 50 00 : 61
7E18 00 04 00 00 00 00 00 00 : 04
7E20 00 00 00 00 00 00 00 00 : 00
7E28 00 00 00 00 70 00 00 04 : 74
7E30 00 00 00 20 00 00 00 00 : 20
7E38 00 00 00 00 00 00 00 00 : 00
7E40 00 00 60 00 05 00 00 00 : 65
7E48 00 00 00 00 00 00 00 00 : 00
7E50 00 00 00 00 00 00 00 00 : 00
7E58 00 00 00 00 00 00 00 00 : 00
7E60 00 00 00 00 00 00 00 00 : 00
7E68 00 00 00 00 00 00 00 00 : 00
7E70 00 00 00 00 00 00 00 00 : 00
7E78 00 00 40 00 00 00 00 00 : 40
---------------------------------
SUM: 22 15 B1 31 86 22 61 15 ADD2

7E80 00 00 00 00 00 00 00 00 : 00
7E88 00 00 00 00 00 00 00 00 : 00
7E90 04 00 00 00 00 00 00 00 : 04
7E98 00 00 00 00 00 00 00 00 : 00
7EA0 00 00 00 00 00 00 40 20 : 60
7EA8 00 00 00 00 00 00 00 00 : 00
7EB0 00 00 00 00 00 00 00 00 : 00
7EB8 00 00 00 00 11 10 00 00 : 21
7EC0 00 00 00 00 00 00 00 00 : 00
7EC8 00 00 00 00 00 00 00 00 : 00
7ED0 00 00 00 00 00 00 00 00 : 00
7ED8 00 00 00 00 00 00 00 00 : 00
7EE0 00 00 00 00 00 00 00 00 : 00
7EE8 00 00 00 00 00 00 00 00 : 00
7EF0 00 00 00 00 00 00 00 00 : 00
7EF8 00 00 00 00 00 00 00 00 : 00
---------------------------------
SUM: 04 00 00 00 11 10 40 20 8D65

7F00 00 00 00 00 00 00 00 00 : 00
7F08 00 00 00 00 00 00 00 00 : 00
7F10 00 00 00 00 00 00 00 00 : 00
7F18 00 00 00 00 00 00 50 00 : 50
7F20 06 00 00 00 00 00 00 00 : 06
7F28 00 00 00 00 00 00 00 00 : 00
7F30 00                      : 00
---------------------------------
SUM: 06 00 00 00 00 00 50 00 B06E
```

リスト3 CUBE_TERU.CBS

```
7000 0F 4D 65 72 63 75 72 79 : F6
7008 20 20 20 20 53 74 61 67 : 0F
7010 65 00 06 00 00 00 00 00 : 6B
7018 00 15 00 00 00 00 00 03 : 18
7020 00 00 00 01 10 00 00 00 : 11
7028 03 00 00 00 00 00 01 10 : 14
7030 00 00 00 00 00 04 00 00 : 04
7038 00 00 14 00 00 00 00 00 : 14
7040 00 00 00 00 00 10 00 00 : 10
7048 00 00 00 00 00 00 00 11 : 11
7050 10 00 00 11 11 11 11 10 : 64
7058 00 00 11 10 00 00 00 00 : 21
7060 00 00 00 00 00 00 00 00 : 00
7068 00 00 00 00 00 00 00 00 : 00
7070 00 00 00 00 00 00 00 00 : 00
7078 00 00 00 00 00 00 00 00 : 00
---------------------------------
SUM: A7 82 B0 B4 D7 0E E5 14 25D0

7080 01 11 00 00 00 00 00 00 : 12
```

```
7088 00 00 00 01 11 00 00 00 : 12
7090 00 02 00 00 00 04 01 00 : 07
7098 00 00 00 00 00 00 00 00 : 00
70A0 00 01 00 00 00 00 00 00 : 01
70A8 00 00 00 00 01 10 00 01 : 12
70B0 11 11 11 11 11 11 00 01 : 67
70B8 10 00 00 00 00 00 00 00 : 10
70C0 00 06 01 00 00 00 00 00 : 07
70C8 00 00 00 00 00 01 00 00 : 01
70D0 00 00 00 00 00 00 00 00 : 00
70D8 01 10 00 00 00 00 00 00 : 11
70E0 00 00 00 01 10 00 00 00 : 11
70E8 00 00 00 00 00 00 01 00 : 01
70F0 00 00 07 00 00 00 70 00 : 77
70F8 00 01 00 00 00 00 00 00 : 01
---------------------------------
SUM: 23 3C 19 13 33 26 72 02 DDE5

7100 00 00 00 00 00 00 00 02 : 02
7108 00 00 00 00 00 00 00 00 : 00
```

```
7110 00 00 00 00 00 00 00 00 : 00
7118 00 00 00 00 00 00 00 00 : 00
7120 00 00 00 00 00 00 00 00 : 00
7128 00 00 00 00 00 00 00 00 : 00
7130 00 00 00 00 00 00 00 00 : 00
7138 00 00 00 00 00 00 00 00 : 00
7140 00 00 00 00 00 00 00 00 : 00
7148 30 00 00 03 00 00 00 00 : 33
7150 00 00 00 00 00 00 00 00 : 00
7158 00 00 00 00 00 03 00 00 : 03
7160 00 00 30 00 00 00 00 00 : 30
7168 00 00 00 00 00 00 00 00 : 00
7170 00 00 00 30 00 30 03 00 : 63
7178 03 00 00 00 00 00 00 00 : 03
---------------------------------
SUM: 33 00 30 33 00 33 03 02 B154

7180 00 00 00 00 00 00 00 00 : 00
7188 04 00 03 00 00 30 00 30 : 67
7190 00 00 00 00 00 00 00 00 : 00
```

▶ 卒論の資料はない。就職先は決まってない。HDのバックアップはとってない。PlayStation版「ときめきメモリアル」は予約したが，本体はもっていない。以上，エキサイティングな夏休みでした(？)。

嶋村　謙(22)大阪府

```
7198 00 00 00 00 00 00 70 00 : 70
71A0 70 00 00 03 00 04 00 00 : 77
71A8 00 00 00 00 00 00 00 00 : 00
71B0 00 00 00 00 70 00 40 00 : B0
71B8 00 03 00 06 00 00 00 00 : 09
71C0 00 00 00 00 00 00 00 00 : 00
71C8 00 00 50 00 30 00 00 04 : 84
71D0 00 05 00 00 00 00 00 00 : 05
71D8 00 00 00 00 00 00 00 00 : 00
71E0 30 00 50 00 00 05 00 07 : 8C
71E8 00 00 00 00 00 00 00 00 : 00
71F0 00 00 00 00 00 00 00 00 : 00
71F8 00 00 00 00 00 30 00 00 : 30
----------------------------------
SUM: A4 08 A3 09 A0 69 B0 3B BEBE

7200 00 00 00 00 00 00 00 00 : 00
7208 00 00 00 00 05 00 00 30 : 35
7210 00 00 00 30 00 20 00 00 : 50
7218 00 00 00 00 00 00 00 00 : 00
7220 00 11 11 11 11 11 11 00 : 66
7228 00 11 11 11 11 11 11 11 : 77
7230 11 11 00 00 11 11 11 00 : 55
7238 00 00 01 00 00 00 00 00 : 01
7240 00 00 00 00 00 00 10 00 : 10
7248 00 00 00 00 00 00 40 00 : 40
7250 00 10 03 00 04 00 00 00 : 17
7258 00 00 00 01 00 00 00 00 : 01
7260 00 00 00 11 11 00 11 11 : 44
7268 11 11 11 11 00 00 00 00 : 44
7270 00 00 00 00 00 00 00 00 : 00
7278 00 00 00 00 00 00 00 00 : 00
----------------------------------
SUM: 22 54 37 75 4D 53 94 52 47C1

7280 00 00 00 00 00 00 00 00 : 00
7288 00 00 00 00 00 00 00 00 : 00
7290 50 00 00 00 00 00 00 00 : 50
7298 00 00 00 00 00 00 00 00 : 00
72A0 00 00 00 00 00 00 00 00 : 00
72A8 00 00 00 11 11 00 11 11 : 44
72B0 11 11 11 11 11 00 11 11 : 77
72B8 00 11 11 00 00 00 00 00 : 22
72C0 00 00 00 01 11 11 00 00 : 23
72C8 00 00 00 00 00 00 00 00 : 00
72D0 00 60 00 00 70 00 00 07 : D7
72D8 00 00 06 00 00 00 00 00 : 06
72E0 00 00 00 00 00 00 00 11 : 11
72E8 11 11 11 11 11 11 11 11 : 88
72F0 11 11 11 11 11 11 11 11 : 88
72F8 11 11 11 11 11 11 10 00 : 76
----------------------------------
SUM: 94 B5 5B 56 D6 44 54 5C DA94

7300 00 00 00 00 00 00 00 00 : 00
7308 00 00 00 00 00 00 00 00 : 00
7310 00 00 00 00 00 00 00 00 : 00
7318 00 00 00 00 00 00 00 00 : 00
7320 00 60 00 04 00 00 00 00 : 64
7328 00 00 00 00 00 00 00 00 : 00
7330 00 00 00 00 00 00 00 11 : 11
7338 11 11 11 11 11 11 00 00 : 66
7340 00 00 10 00 60 00 00 07 : 77
7348 00 00 00 00 00 10 00 00 : 10
7350 00 00 00 00 00 00 00 00 : 00
7358 01 00 00 00 00 03 00 00 : 04
7360 00 00 00 01 00 00 00 00 : 01
7368 00 00 00 00 00 00 00 11 : 11
7370 10 00 00 11 11 11 00 40 : 83
7378 00 00 00 00 00 00 00 00 : 00
----------------------------------
SUM: 22 71 21 27 82 35 00 69 0E70

7380 00 00 00 00 00 00 00 50 : 50
7388 00 50 00 00 11 11 11 00 : 83
7390 01 00 00 00 00 00 00 00 : 01
7398 00 00 00 01 00 00 00 00 : 01
73A0 00 00 00 00 00 00 01 11 : 12
73A8 11 11 11 10 00 00 00 00 : 43
73B0 00 00 00 00 00 00 00 00 : 00
73B8 00 00 00 00 00 20 00 00 : 20
73C0 00 00 00 00 00 00 00 00 : 00
73C8 00 00 00 00 00 00 00 00 : 00
73D0 00 00 00 00 00 00 00 00 : 00
73D8 00 00 60 00 00 00 00 00 : 60
73E0 01 10 00 10 00 00 00 00 : 21
73E8 00 00 00 00 00 00 00 00 : 00
73F0 30 00 00 00 00 00 00 00 : 30
73F8 00 00 00 00 00 00 01 11 : 12
----------------------------------
SUM: 43 71 71 21 11 31 13 72 0BC5

7400 11 10 00 11 00 00 30 00 : 62
7408 00 00 11 11 00 00 00 00 : 22
7410 00 00 00 00 00 11 11 00 : 22
7418 00 00 00 00 11 10 00 01 : 22
7420 11 11 00 11 00 00 00 11 : 44
7428 11 11 11 11 11 00 00 00 : 55
7430 00 00 00 00 00 00 00 00 : 00
7438 00 00 00 11 00 00 00 00 : 11
7440 00 00 00 00 00 00 00 00 : 00
7448 00 00 00 11 14 04 00 11 : 3A
7450 10 00 00 00 00 00 01 10 : 21
7458 00 00 11 00 00 00 00 00 : 11
7460 00 00 14 04 00 10 00 00 : 28
7468 00 00 00 00 00 10 00 00 : 10
7470 10 00 00 11 11 11 11 00 : 54
7478 14 04 00 10 01 11 70 70 : 1A
----------------------------------
SUM: 67 36 47 8B 48 67 C3 A3 B9EA

7480 70 70 00 10 00 00 10 70 : 70
7488 70 00 00 00 00 00 14 04 : 88
7490 00 10 00 00 11 11 11 11 : 54
7498 00 10 00 00 10 11 11 00 : 42
74A0 00 00 00 00 14 04 00 10 : 28
74A8 00 00 00 00 00 00 00 10 : 10
74B0 00 00 10 00 00 60 20 00 : 90
74B8 00 00 03 03 06 00 00 00 : 0C
74C0 00 00 00 00 00 00 00 00 : 00
74C8 00 00 00 00 00 40 00 00 : 40
74D0 00 00 00 00 00 00 00 00 : 00
74D8 00 00 00 00 00 00 00 00 : 00
74E0 00 00 00 60 00 00 00 00 : 60
74E8 60 00 00 00 00 00 00 00 : 60
74F0 00 00 00 00 00 00 00 00 : 00
74F8 00 10 10 10 10 10 10 00 : 60
----------------------------------
SUM: 40 A0 23 83 4B D6 76 A5 BD09

7500 00 00 00 00 00 00 04 00 : 04
7508 00 00 00 00 00 00 00 10 : 10
7510 00 00 00 00 00 10 00 00 : 10
7518 00 00 00 00 06 00 00 00 : 06
7520 00 00 00 00 00 10 00 00 : 10
7528 00 00 00 10 00 00 00 00 : 10
7530 00 00 01 01 00 00 00 00 : 02
7538 00 00 00 10 00 00 60 00 : 70
7540 00 10 00 00 00 00 00 00 : 10
7548 01 00 00 00 00 00 00 00 : 01
7550 00 10 00 00 00 00 00 10 : 20
7558 00 00 00 00 00 00 01 00 : 01
7560 00 00 00 00 00 00 00 10 : 10
7568 00 00 00 00 00 10 00 00 : 10
7570 00 00 00 00 01 00 00 60 : 61
7578 00 00 00 00 00 10 00 00 : 10
----------------------------------
SUM: 01 20 01 21 07 40 65 90 108E

7580 00 00 00 00 00 00 00 00 : 00
7588 00 00 01 01 01 01 00 00 : 04
7590 00 00 00 10 00 00 00 00 : 10
7598 00 00 00 00 00 00 10 00 : 10
75A0 00 00 00 00 00 00 00 20 : 20
75A8 00 00 00 00 00 00 00 00 : 00
75B0 00 00 00 00 00 00 00 00 : 00
75B8 00 00 00 00 00 00 00 00 : 00
75C0 00 00 00 00 00 00 00 00 : 00
75C8 00 00 00 00 00 00 00 00 : 00
75D0 00 00 00 05 01 11 11 11 : 39
75D8 11 11 11 11 10 00 00 00 : 54
75E0 00 01 00 00 00 10 00 00 : 11
75E8 00 00 40 00 01 11 11 11 : 74
75F0 10 00 00 00 00 00 00 01 : 11
75F8 00 00 00 10 00 00 00 00 : 10
----------------------------------
SUM: 21 12 52 37 13 33 32 43 A688

7600 05 02 01 11 11 11 10 05 : 50
7608 00 00 00 00 00 01 00 00 : 01
7610 00 10 00 00 00 00 01 11 : 22
7618 11 11 11 11 11 11 10 00 : 76
7620 00 01 00 00 00 30 00 00 : 31
7628 00 10 00 00 01 00 00 00 : 11
7630 00 00 00 00 10 00 00 01 : 11
7638 00 00 11 11 11 00 00 10 : 43
7640 00 00 00 00 00 00 40 00 : 40
7648 00 40 00 00 04 00 00 00 : 44
7650 00 00 00 00 00 00 00 00 : 00
7658 00 10 00 01 11 11 11 15 : 59
7660 00 00 04 01 11 10 00 00 : 26
7668 11 00 00 00 00 00 00 11 : 22
7670 00 01 11 11 11 13 04 00 : 4B
7678 04 01 10 10 00 10 00 01 : 36
----------------------------------
SUM: 2B 86 48 56 7B 97 76 4E 4BD9

7680 00 00 00 00 01 10 10 01 : 22
7688 11 11 11 13 04 00 00 00 : 4A
7690 10 10 11 00 00 00 10 00 : 41
7698 00 05 00 10 01 01 00 30 : 47
76A0 00 11 15 00 00 00 10 00 : 36
76A8 01 00 00 00 10 00 00 00 : 11
76B0 00 00 00 00 00 00 00 00 : 00
76B8 00 00 00 00 00 00 00 00 : 00
76C0 00 00 00 00 00 00 00 07 : 07
76C8 00 00 00 00 50 00 00 00 : 50
76D0 00 00 00 00 00 00 00 00 : 00
76D8 00 00 00 11 11 11 10 00 : 43
76E0 00 00 70 00 00 00 00 60 : D0
76E8 00 00 00 00 00 00 00 00 : 00
76F0 00 00 00 00 00 00 00 00 : 00
76F8 70 00 70 00 07 00 00 00 : E7
----------------------------------
SUM: 92 37 17 34 7E 22 40 98 C7BF

7700 00 00 00 00 00 00 00 00 : 00
7708 00 00 00 04 00 00 11 11 : 26
7710 11 11 00 00 00 00 00 00 : 22
7718 00 00 00 00 00 00 00 00 : 00
7720 40 04 00 40 00 00 00 00 : 84
7728 00 10 01 00 10 01 00 10 : 32
7730 01 00 10 00 14 01 40 10 : 76
7738 01 40 10 01 00 10 00 10 : 72
7740 01 00 10 01 00 10 01 00 : 23
7748 10 40 14 01 00 10 01 00 : 76
7750 10 01 30 10 00 10 01 00 : 62
7758 10 01 00 10 01 00 10 40 : 72
7760 10 01 00 10 01 00 10 01 : 33
7768 30 10 00 10 01 00 10 01 : 62
7770 00 10 01 00 10 00 10 01 : 32
7778 00 10 01 00 10 01 00 10 : 32
----------------------------------
SUM: C4 D8 77 87 47 43 94 94 23F8

7780 00 10 01 00 10 01 00 10 : 32
7788 01 00 10 70 16 01 70 15 : 1D
7790 01 70 12 01 70 10 00 10 : 14
7798 01 00 10 01 00 10 01 00 : 23
77A0 10 00 00 00 00 00 00 00 : 10
77A8 00 00 00 00 00 00 00 00 : 00
77B0 00 70 00 00 00 00 00 00 : 70
77B8 00 00 00 00 00 00 00 00 : 00
77C0 00 00 00 00 00 00 00 60 : 60
77C8 00 02 00 00 00 00 00 00 : 02
77D0 00 00 00 00 00 00 00 00 : 00
77D8 00 00 11 11 11 11 11 11 : 66
77E0 11 11 00 00 00 00 00 00 : 22
77E8 00 00 00 00 00 00 00 00 : 00
77F0 00 00 00 00 00 00 00 00 : 00
77F8 00 00 00 00 40 00 04 00 : 44
----------------------------------
SUM: 24 03 44 83 E7 33 86 A6 3B5D

7800 00 00 00 00 00 00 00 00 : 00
7808 00 00 00 00 00 00 00 11 : 11
7810 11 11 11 11 11 11 00 00 : 66
7818 00 00 00 00 00 00 00 00 : 00
7820 00 00 00 00 00 00 00 00 : 00
7828 00 00 00 00 00 00 00 00 : 00
7830 00 00 00 00 00 00 00 04 : 04
7838 00 00 00 00 00 40 00 00 : 40
7840 00 00 00 00 00 00 00 00 : 00
7848 00 00 00 00 11 11 11 10 : 43
7850 00 00 00 30 00 00 00 00 : 30
7858 00 00 00 00 00 00 00 00 : 00
7860 00 00 00 00 00 00 00 11 : 11
7868 11 11 11 11 10 00 00 00 : 54
7870 10 40 00 00 00 00 00 00 : 50
7878 00 00 00 10 00 70 00 00 : 80
----------------------------------
SUM: 32 62 22 62 32 D2 11 36 EA43

7880 00 00 00 00 00 00 10 60 : 70
7888 00 00 00 00 00 00 00 00 : 00
7890 00 10 00 00 00 00 00 00 : 10
7898 00 00 00 00 00 00 00 00 : 00
78A0 00 00 00 00 00 00 00 00 : 00
78A8 00 00 00 00 30 00 00 00 : 30
78B0 00 04 00 00 00 00 00 00 : 04
78B8 00 00 00 00 00 00 00 00 : 00
78C0 00 00 20 00 00 00 00 05 : 25
78C8 00 00 00 00 00 00 00 00 : 00
78D0 00 00 00 00 00 00 00 00 : 00
78D8 30 00 00 00 00 06 00 00 : 36
78E0 00 00 00 00 00 00 00 00 : 00
78E8 00 00 00 00 00 00 70 00 : 70
78F0 00 00 00 06 00 00 00 00 : 06
78F8 00 00 00 00 00 00 00 00 : 00
----------------------------------
SUM: 30 14 20 06 30 06 80 65 EAFC

7900 00 00 00 00 60 00 00 00 : 60
7908 00 07 00 00 00 00 00 00 : 07
7910 00 00 00 00 00 00 00 00 : 00
7918 00 00 30 00 00 00 00 07 : 37
7920 00 00 00 00 00 00 00 00 : 00
7928 00 00 00 00 00 00 00 11 : 11
7930 11 00 00 00 00 11 11 11 : 44
7938 00 00 11 11 00 00 00 00 : 22
7940 11 11 11 00 00 11 11 00 : 55
7948 00 00 00 11 11 11 00 00 : 33
7950 11 11 11 11 11 11 11 11 : 88
7958 11 00 00 00 00 00 00 00 : 11
7960 00 00 00 00 00 00 00 00 : 00
7968 00 00 00 00 00 00 00 00 : 00
7970 00 00 00 00 00 00 11 00 : 11
7978 40 50 00 00 00 00 00 00 : 90
----------------------------------
SUM: 84 79 63 33 82 44 44 3A C6EF

7980 00 11 00 00 00 00 02 00 : 13
7988 00 00 00 00 00 00 00 00 : 00
7990 00 00 00 04 00 00 00 00 : 04
7998 00 00 00 00 00 00 00 00 : 00
79A0 00 00 00 00 50 00 00 00 : 50
79A8 00 03 00 00 30 00 00 00 : 33
79B0 00 00 00 00 00 00 00 00 : 00
79B8 00 00 11 11 00 11 11 11 : 55
79C0 11 11 11 11 00 00 00 00 : 44
79C8 10 03 00 01 11 11 11 00 : 47
79D0 30 30 00 15 00 00 10 00 : 85
```

▶私はCD-ROMドライブをもっている。解析という遊びは結構面白い。しかし，Windows用のCD-ROMの解析にはWindowsマシンが必要だと思うのだが，皆さんはどうやっているのだろう。
西山　新志(24)福岡県

```
79D8 00 00 00 00 00 00 10 00 : 10
79E0 11 10 00 00 00 00 30 30 : 81
79E8 00 16 01 00 00 00 70 00 : 87
79F0 00 00 00 00 10 01 00 00 : 11
79F8 00 00 00 00 30 30 00 11 : 71
---------------------------------
SUM: 62 7E 23 3C D1 53 E4 52 A6A4

7A00 11 00 00 00 11 00 00 00 : 22
7A08 00 00 00 00 00 00 00 11 : 11
7A10 00 00 70 30 00 00 00 00 : A0
7A18 00 11 11 00 00 00 00 00 : 22
7A20 00 00 00 00 11 11 00 00 : 22
7A28 30 30 00 11 11 11 00 11 : A4
7A30 11 00 00 00 00 00 11 11 : 33
7A38 11 00 00 00 00 00 30 30 : 71
7A40 00 00 11 11 00 00 00 00 : 22
7A48 00 00 00 00 00 11 11 00 : 22
7A50 00 00 00 00 30 30 00 00 : 60
7A58 11 11 11 11 00 00 00 00 : 44
7A60 00 60 00 11 11 11 11 11 : B5
7A68 11 11 30 30 00 00 11 11 : A4
7A70 00 00 00 00 00 00 00 00 : 00
7A78 00 00 00 00 20 00 00 00 : 20
---------------------------------
SUM: 85 C3 D3 A4 94 74 74 85 FF59

7A80 00 00 00 00 00 00 00 00 : 00
7A88 00 00 00 00 00 00 00 00 : 00
7A90 00 00 40 00 00 00 00 00 : 40
7A98 00 00 00 00 00 00 00 00 : 00
7AA0 00 00 00 00 00 00 00 40 : 40
7AA8 40 40 00 00 00 00 00 00 : 80
7AB0 00 00 00 00 00 00 00 00 : 00
7AB8 00 00 00 00 70 40 40 40 : 30
7AC0 60 00 00 00 00 00 00 00 : 60
7AC8 00 00 00 00 00 00 00 00 : 00
7AD0 00 00 30 40 40 40 30 00 : 20
7AD8 00 00 00 00 00 00 00 00 : 00
7AE0 00 00 00 00 00 00 00 00 : 00
7AE8 30 40 40 40 30 00 00 00 : 20
7AF0 00 00 00 00 00 00 00 00 : 00
7AF8 00 00 00 00 00 00 60 40 : A0
---------------------------------
SUM: D0 80 B0 80 E0 80 D0 C0 5BB4

7B00 40 70 70 00 00 00 00 00 : 20
7B08 00 00 00 00 00 00 00 00 : 00
7B10 00 00 00 00 30 30 50 30 : E0
7B18 30 00 00 00 00 00 00 00 : 30
7B20 00 00 00 00 00 00 00 00 : 00
7B28 00 00 50 60 30 40 50 00 : 70
7B30 00 00 00 00 00 00 00 00 : 00
7B38 00 00 00 00 00 00 00 00 : 00
7B40 70 40 30 30 60 00 00 00 : 70
7B48 00 00 00 00 00 00 00 00 : 00
7B50 00 00 00 00 07 00 40 40 : 87
```

```
7B58 70 30 30 07 00 00 00 00 : D7
7B60 00 00 00 00 00 00 00 00 : 00
7B68 00 00 00 00 00 00 00 00 : 00
7B70 00 00 00 00 00 00 00 00 : 00
7B78 00 00 00 00 00 00 00 02 : 02
---------------------------------
SUM: 50 E0 20 97 C7 70 E0 72 BA49

7B80 00 00 07 00 00 00 00 00 : 07
7B88 00 00 00 00 00 00 00 00 : 00
7B90 00 00 00 00 00 11 11 11 : 33
7B98 11 11 11 11 10 00 00 00 : 54
7BA0 00 00 00 01 11 11 11 10 : 44
7BA8 00 00 00 00 00 00 00 00 : 00
7BB0 00 00 00 00 00 00 00 00 : 00
7BB8 00 00 00 00 00 00 00 00 : 00
7BC0 00 00 00 04 00 04 00 03 : 0B
7BC8 00 00 00 00 00 00 00 00 : 00
7BD0 00 00 00 00 00 00 00 11 : 11
7BD8 00 11 11 11 11 11 11 11 : 77
7BE0 00 00 11 00 11 11 11 11 : 55
7BE8 11 11 11 00 00 70 00 11 : B4
7BF0 11 11 00 00 00 00 00 00 : 22
7BF8 00 00 00 00 30 00 00 00 : 30
---------------------------------
SUM: 33 44 4B 27 73 B8 44 68 4276

7C00 00 00 00 11 00 00 00 00 : 11
7C08 00 01 11 11 11 11 11 00 : 56
7C10 11 11 11 00 01 11 11 11 : 67
7C18 11 70 00 00 00 30 00 00 : B1
7C20 01 11 11 11 00 00 00 00 : 34
7C28 00 00 00 00 11 11 11 11 : 44
7C30 00 11 11 11 00 00 00 01 : 34
7C38 11 11 11 00 00 00 00 00 : 33
7C40 00 00 00 11 11 11 00 00 : 33
7C48 00 00 00 00 00 00 01 11 : 12
7C50 11 00 00 00 00 00 00 00 : 11
7C58 00 00 11 00 00 00 00 00 : 11
7C60 20 00 00 00 00 00 00 00 : 20
7C68 00 00 00 00 00 00 00 00 : 00
7C70 00 00 70 00 03 00 00 03 : 76
7C78 00 00 60 00 00 00 00 00 : 60
---------------------------------
SUM: 65 B5 36 55 37 74 34 37 99D7

7C80 00 00 00 00 00 00 00 11 : 11
7C88 11 00 01 11 11 11 11 00 : 56
7C90 11 11 11 11 00 01 11 11 : 67
7C98 11 11 00 11 11 00 00 00 : 44
7CA0 00 00 30 00 00 00 00 00 : 30
7CA8 00 00 00 00 00 00 00 00 : 00
7CB0 00 00 00 00 40 00 00 00 : 40
7CB8 30 00 00 00 30 00 00 00 : 60
7CC0 00 00 00 00 00 00 00 00 : 00
7CC8 00 11 11 11 10 01 11 00 : 55
7CD0 11 11 11 00 11 11 11 10 : 76
```

```
7CD8 01 11 00 11 11 11 00 00 : 45
7CE0 00 04 00 00 00 00 40 00 : 44
7CE8 00 00 00 00 00 00 00 00 : 00
7CF0 00 00 00 00 00 00 00 00 : 00
7CF8 40 00 00 04 00 00 00 00 : 44
---------------------------------
SUM: B5 59 64 59 C4 35 84 32 CA5B

7D00 00 00 00 00 00 00 00 00 : 00
7D08 00 00 00 11 00 11 11 10 : 43
7D10 01 11 11 00 11 11 11 00 : 56
7D18 11 11 10 01 11 11 00 11 : 66
7D20 11 00 00 00 00 00 00 00 : 11
7D28 00 00 00 00 00 00 00 00 : 00
7D30 00 00 00 00 00 00 00 00 : 00
7D38 00 06 00 00 00 07 00 00 : 0D
7D40 00 00 00 00 00 00 00 00 : 00
7D48 00 00 00 00 00 00 00 00 : 00
7D50 00 06 00 00 00 00 00 00 : 06
7D58 00 00 00 00 00 00 00 00 : 00
7D60 00 00 00 00 00 00 00 05 : 05
7D68 00 00 00 00 00 00 00 00 : 00
7D70 00 00 00 00 00 00 00 00 : 00
7D78 00 00 00 11 11 11 11 11 : 55
---------------------------------
SUM: 23 2E 21 23 33 4B 33 37 7095

7D80 11 11 00 00 00 00 11 11 : 44
7D88 11 11 11 11 11 00 00 00 : 55
7D90 00 00 00 00 00 00 00 00 : 00
7D98 00 00 00 00 00 00 00 00 : 00
7DA0 00 00 00 00 00 00 50 00 : 50
7DA8 00 00 00 00 00 00 60 00 : 60
7DB0 00 00 00 00 00 00 00 00 : 00
7DB8 00 00 00 00 40 00 00 00 : 40
7DC0 00 00 00 00 40 00 00 00 : 40
7DC8 00 00 00 00 00 00 00 00 : 00
7DD0 00 01 10 00 01 10 00 01 : 23
7DD8 10 00 01 10 01 10 00 01 : 33
7DE0 10 00 01 10 00 01 10 01 : 33
7DE8 11 11 11 10 00 01 11 11 : 66
7DF0 11 10 01 11 11 11 10 00 : 65
7DF8 01 11 11 11 10 00 00 00 : 44
---------------------------------
SUM: 65 55 46 63 B4 33 F2 25 8931

7E00 00 00 20 00 00 00 00 00 : 20
7E08 00 00 00 00 00 00 00 00 : 00
7E10 00 00 00 00 00 11 11 11 : 33
7E18 11 11 11 11 10 00 00 00 : 54
7E20 01 11 11 11 11 11 11 00 : 67
7E28 00 00 50 00 00 00 00 00 : 50
7E30 00 00 06 00 00 00 00 00 : 06
7E38 00 00 00 00 00 00 00    : 00
---------------------------------
SUM: 12 22 98 22 21 22 22 11 5E2E
```

▶ 全機種共通システムインデックス ◀

＊以下のアプリケーションは，基本システムであるS-OS“MACE”またはS-OS“SWORD”がないと作動しませんのでご注意ください。

1985

■**85年6月号**
序論　共通化の試み
第1部　S-OS“MACE”
第2部　Lisp-85インタプリタ
第3部　チェックサムプログラム
■**85年7月号**
第4部　マシン語プログラム開発入門
第5部　エディタアセンブラZEDA
第6部　デバグツールZAID
■**85年8月号**
第7部　ゲーム開発パッケージBEMS
第8部　ソースジェネレータZING
■**85年9月号**
インタラプト　S-OS番外地
第9部　マシン語入力ツールMACINTO-S
第10部　Lisp-85入門(1)
■**85年10月号**
第11部　仮想マシンCAP-X85
連載　Lisp-85入門(2)
■**85年11月号**
連載　Lisp-85入門(3)
■**85年12月号**
第12部　Prolog-85発表

1986

■**86年1月号**
第13部　リロケータブルのお話
第14部　FM音源サウンドエディタ
■**86年2月号**
第15部　S-OS“SWORD”
第16部　Prolog-85入門(1)
■**86年3月号**
第17部　magiFORTH発表
連載　Prolog-85入門(1)
■**86年4月号**
第18部　思考ゲームJEWEL
第19部　LIFE GAME
連載　基礎からのmagiFORTH
連載　Prolog-85入門(3)
■**86年5月号**
第20部　スクリーンエディタE-MATE
連載　実戦演習magiFORTH
■**86年6月号**
第21部　Z80TRACER
第22部　magiFORTH TRACER
第23部　ディスクダンプ＆エディタ
第24部　“SWORD”2000 QD
連載　対話で学ぶmagiFORTH

特別付録　PC-8801版S-OS“SWORD”
■**86年7月号**
第25部　FM音源ミュージックシステム
付録　FM音源ボードの製作
連載　計算力アップのmagiFORTH
特別付録　SMC-777版S-OS“SWORD”
■**86年8月号**
第26部　対局五目並べ
第27部　MZ-2500版S-OS“SWORD”
■**86年9月号**
第28部　FuzzyBASIC発表
連載　明日に向かってmagiFORTH
■**86年10月号**
第29部　ちょっと便利な拡張プログラム
第30部　ディスクモニタDREAM
第31部　FuzzyBASIC料理法＜1＞
■**86年11月号**
第32部　パズルゲームHOTTAN
第33部　MAZE in MAZE
連載　FuzzyBASIC料理法＜2＞
■**86年12月号**
第34部　CASL & COMET
連載　FuzzyBASIC料理法＜3＞

▶今年の夏はゴキ○リ叩きが熱い！　しかし本体の裏，寝床のすぐ脇で逃げられてからは……。そのうえ，雷とブレーカーダウンの嵐。もはや……(現実逃避モード)。
西川　和範(20)東京都

1987

■87年1月号
第35部　マシン語入力ツールMACINTO-C
連載　FuzzyBASIC料理法＜4＞
■87年2月号
第36部　アドベンチャーゲームMARMALADE
第37部　テキアベ作成ツールCONTEX
■87年3月号
第38部　魔法使いはアニメがお好き
第39部　アニメーションツールMAGE
付録　“SWORD”再掲載とMAGICの標準化
■87年4月号
第40部　INVADER GAME
第41部　TANGERINE
■87年5月号
第42部　S-OS“SWORD”変身セット
第43部　MZ-700“SWORD”をQD対応に
■87年6月号
インタラプト　コンパイラ物語
第44部　FuzzyBASICコンパイラ
第45部　エディタアセンブラZEDA-3
■87年7月号
第46部　STORY MASTER
■87年8月号
第47部　パズルゲーム碁石拾い
第48部　漢字出力パッケージJACKWRITE
特別付録　FM-7/77版S-OS“SWORD”
■87年9月号
第49部　リロケータブル逆アセンブラInside-R
特別付録　PC-8001/8801版S-OS“SWORD”
■87年10月号
第50部　tiny CORE WARS
第51部　FuzzyBASICコンパイラの拡張
第52部　XIturob版S-OS“SWORD”
■87年11月号
序論　神話のなかのマイクロコンピュータ
付録　S-OSの仲間たち
第53部　もうひとつのFuzzyBASIC入門
第54部　ファイルアロケータ＆ローダ
インタラプト　S-OSこちら集中治療室
第55部　BACK GAMMON
■87年12月号
第56部　タートルグラフィックパッケージTURTLE
第57部　XIturbo版“SWORD”アフターケア
ラインプリントルーチン
特別付録　PASOPIA7版S-OS“SWORD”

1988

■88年1月号
第58部　FuzzyBASICコンパイラ・奥村版
付録　石上版コンパイラ拡張部の修正
■88年2月号
第59部　シューティングゲームELFES
■88年3月号
第60部　構造型コンパイラ言語SLANG
■88年4月号
第61部　デバッギングツールTRADE
第62部　シミュレーションウォーゲームWALRUS
■88年5月号
第63部　シューティングゲームELFESII
第64部　地底最大の作戦
■88年6月号
第65部　構造化言語SLANG入門(1)
第66部　Lisp-85用NAMPAシミュレーション
■88年7月号
第67部　マルチウィンドウドライバーMW-1
連載　構造化言語SLANG入門(2)
■88年8月号
第68部　マルチウィンドウエディタWINER
■88年9月号
第69部　超小型エディタTED-750
第70部　アフターケアWINERの拡張
■88年10月号
第71部　SLANG用ファイル入出力ライブラリ
第72部　シューティングゲームMANKAI
■88年11月号
第73部　シューティングゲームFLFESIV
■88年12月号
第74部　ソースジェネレータSOURCERY

1989

■89年1月号
第75部　パズルゲームLAST ONE
第76部　ブロックゲームFLICK
■89年2月号
第77部　高速エディタアセンブラREDA
特別付録　XI版S-OS“SWORD”＜再掲載＞
■89年3月号
第78部　Z80用浮動小数点演算パッケージSOR
OBAN
■89年4月号
第79部　SLANG用実数演算ライブラリ
■89年5月号
第80部　ソースジェネレータRING
■89年6月号
第81部　超小型コンパイラTTC
■89年7月号
第82部　TTC用パズルゲームTICBAN
■89年8月号
第83部　CP/M用ファイルコンバータ
■89年9月号
第84部　生物進化シミュレーションBUGS
■89年10月号
第85部　小型インタプリタ言語TTI
■89年11月号
第86部　TTI用パズルゲームPUSH BON!
■89年12月号
第87部　SLANG用リダイレクションライブラリDIO.LIB

1990

■90年1月号
第88部　SLANG用ゲームWORM KUN
特別付録　再掲載SLANGコンパイラ
■90年2月号
第89部　超小型コンパイラTTC＋＋
■90年3月号
第90部　超多機能アセンブラOHM-Z80
■90年4月号
第91部　ファジィコンピュータシミュレーションI-MY
■90年5月号
第92部　インタプリタ言語STACK
■90年6月号
第93部　リロケータブルフォーマットの取り決め
第94部　STACK用ゲームSQUASH!
第95部　X68000対応S-OS“SWORD”
特別付録　PC-286対応S-OS“SWORD”
■90年7月号
第96部　リロケータブルアセンブラWZD
■90年8月号
第97部　リンカWLK
■90年9月号
第98部　BILLIARDS
■90年10月号
第99部　ライブラリアンWLB
■90年11月号
第100部　タブコート対応エディタEDC-T
■90年12月号
第101部　STACKコンパイラ

1991

■91年1月号
第102部　ブロックアクションゲームCOLUMNS
■91年2月号
第103部　ダイスゲームKISMET
■91年3月号
第104部　アクションゲームMUD BALLIN'
■91年4月号
第105部　SLANG用カードゲームDOBON
■91年5月号
第106部　実数型コンパイラ言語REAL
■91年6月号
第107部　Small-C処理系の移植
■91年7月号
第108部　REALソースリスト編
■91年8月号
第109部　Small-Cライブラリの移植
■91年9月号
第110部　SLANG用NEWファイル出力ライブラリ
■91年10月号
第111部　Small-C活用講座（初級編）
■91年11月号
第112部　Small-C活用講座（応用編）
第113部　MORTAL
■91年12月号
第114部　Small-C SLANGコンパチ関数

1992

■92年1月号
第115部　LINER
■92年2月号
第116部　シミュレーションゲームPOLANYI
■92年3月号
第117部　カードゲームKLONDIKE
■92年4月号
第118部　オプティマイザO80実践Small-C講座(1)
■92年5月号
第119部　COMMAND.OBJ実践Small-C講座(2)
■92年6月号
第120部　COMMAND.OBJ2実践Small-C講座(3)
■92年7月号
第121部　関数リファレンス実践Small-C講座(4)
■92年8月号
第122部　ワイルドカード実践Small-C講座(5)
第123部　グラフィックライブラリ GRATH.LIB
■92年9月号
第124部　O-EDIT&MODCNV
■92年10月号
第125部　SLENDER HUL実践Small-C講座(6)
■92年11月号
第126部　EDIT実線Small-C講座(7)
■92年12月号
第127部　MAKE実践Small-C講座(8)

1993

■93年1月号
第128部　EDC-Tの拡張
■93年2月号
第129部　BLACK JACK
■93年3月号
第130部　シューティングゲームコアシステム作成法(1)
■93年4月号
第131部　シューティングゲームコアシステム作成法(2)
■93年5月号
第132部　シューティングゲームコアシステム作成法(3)
■93年6月号
第133部　REVERSI
■93年7月号
特別付録　MSX用S-OS“SWORD”
■93年8月号
第134部　MACINTO-C再掲載
■93年9月号
第135部　7並べ
特別付録　SLANG再々掲載
■93年10月号
第136部　シューティングゲームコアシステム作成法(4)
■93年11月号
第137部　S-OSで学ぶZ80マシン語講座(1)
■93年12月号
第138部　エディタアセンブラREDA再掲載

1994

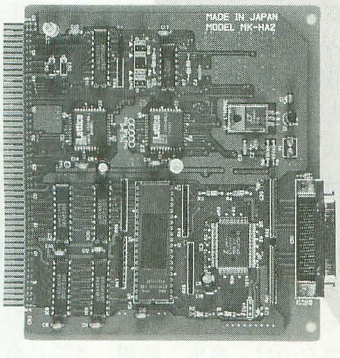
■94年1月号
第139部　S-OSで学ぶZ80マシン語講座(2)
■94年2月号
第140部　YGCSver.0.20ユーザーマニュアル
第141部　S-OSで学ぶZ80マシン語講座(3)
■94年3月号
第142部　S-OSで学ぶZ80マシン語講座(4)
■94年4月号
第143部　S-OSで学ぶZ80マシン語講座(5)
■94年5月号
第144部　S-OSで学ぶZ80マシン語講座(6)
■94年6月号
第145部　YGCSver.0.30
■94年7月号
第146部　シューティングゲーム作成講座(1)
■94年8月号
第147部　シューティングゲーム作成講座(2)
■94年9月号
第148部　怪しいZ80の使い方(テクニック編)
■94年10月号
第149部　シューティングゲーム作成講座(3)
第150部　怪しいZ80の使い方(未定義命令編)
■94年11月号
第151部　B-GALET2
■94年12月号
第152部　シューティングゲーム作成講座(4)

愛読者モニタ

A Mach-2

1名

9月号で大活躍のSCSI-2ボード「Mach-2」。SCSI転送速度を大幅にアップしてくれるぞ。



満開製作所　☎03(3554)9282

モニタの応募方法

希望するモニタ記号をとじ込みのアンケートハガキの左下のスペースまたは官製ハガキに記入してお申し込みください。応募の際に使用環境を明記する必要はありませんが，当選された方にはモニタとして使用ののちレポートを提出していただきます。締め切りは1995年10月18日の到着分までとし，当選者の発表は1995年12月号で行います。また，雑誌公正競争規約の定めにより，当選された方はこの号のほかの懸賞に当選できない場合がありますので，ご了承ください。

B ディスプレイジャック　各1名

1）MK-RGB21-15
2）MK-RGB21-15S

RGB21ピン出力のみのゲーム機をRGB15ピンに変換してくれるアダプタ。きれいな画面でゲームを楽しみたい人は必携の周辺機器です。



満開製作所　☎03(3554)9282

愛読者プレゼント

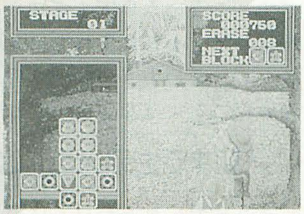
プレゼントの応募方法

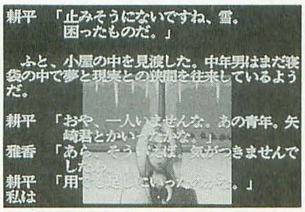
とじ込みのアンケートハガキの該当項目をすべてご記入のうえ，希望するプレゼント番号をハガキ右下のスペースにひとつ記入してお申し込みください。締め切りは1995年10月18日の到着分までとします。当選者の発表は1995年12月号で行います。また，雑誌公正競争規約の定めにより，当選された方はこの号のほかの懸賞に当選できない場合がありますので，ご了承ください。

1 同人ソフトセット　3名

ほいっぷるX68k	X68000用	5″/3.5″2HD版	1,000円(税込)
Devil's letter of Invitation	X68000用	5″/3.5″2HD版	1,700円(税込)

ちょっと変わった押し上げパズルゲームと小説感覚のサスペンスアドベンチャーを楽しく遊ぼう。



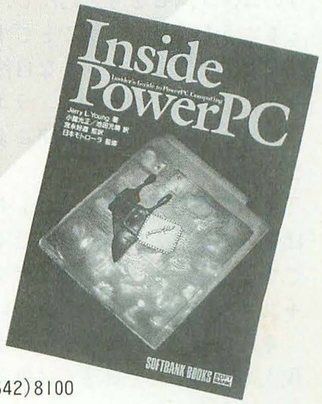


TAKERU事務局　☎052(824)2493

2 Inside PowerPC

3名

マニュアルだけではわからない，PowerPCの思想や時代背景を知ることかできるお勧めの1冊です。



ソフトバンク　☎03(5642)8100

8月号モニタ当選者

[A]マルチメディア・パワード・スピーカー＆スーパーウーファ・システム　(神奈川県)永井　孝　(埼玉県)島田　貴之

8月号プレゼント当選者

[1]バラデューク　(大阪府)溝田　幸弘　(宮城県)小野寺　健一　(群馬県)久保田　智久　[2]マーブルマッドネス　(神奈川県)佐藤　芳臣　(福岡県)島崎　智博　(大阪府)折坂　信春　[3]三國志IIIハンドブック　(埼玉県)布施　泰雄　(大阪府)福森　淳　(敬称略)
以上の方々が当選しました。商品は順次発送致しますが，入荷状況などにより遅れる場合もあります

テレビから逃れた海岸で

キャンプ場のテレビを憎む

最近，にわかにアウトドアづいてきまして，時間をみつけては海や山などにテントをもって出かけています。夏の間が主ですけれどね。で，キャンプをする際，どこにテントを張るかということは自分にとって大きな問題なのです。

リッチなオートキャンパーの方々とは，おのずと趣味も異なってくるものでして，率直にいって，格好いいキャンピングカーで乗りつけて，設備の整ったキャンプ場にぎっしりと詰め込まれ，キャンプグッズを自慢しあうといった風景はうんざりです。

別に，意識して自慢しているわけではないのでしょうが，どちらを見てもお隣のキャンプ生活しか見えぬわけですから，結局，わざわざ遠くまで車で乗りつけ，自然のなかでいかに普段の日常生活を再現できるかというところがポイントにならざるをえないのではないかと思えてしまうのです。

なかでも，これだけはやめてほしいと思うのは，テレビをもち込んで，辺りに聞こえるような音量で流すことです。寝るときに聞こえてくるさまざまな自然の音を楽しみにしているというのに，テレビのCMだとかのくだらない音声が聞こえてくるのはこたえます。

「キャンプ場にはテレビはもち込むべきではないのだ」と怒鳴り込みたくなる気持ちです。ですから，最近は「ゲリラ」的にキャンプ場以外でキャンプしています。

それにしても，テレビというものは，現代人の心のなんと奥深くに巣食ってしまったのでしょうか？

そう，僕は最近テレビには飽き飽きしているのです。なぜなら，僕にとって面白い番組が数少ないからです。

司会者のひやっとする発言

実際，テレビ(元祖X1のモニタで本体はもう大昔に成仏しているが，案外大切に使っている)は毎月1回の粗大ごみに出したほうがいいのでしょう。でも，まだ修行が足りないらしく，それができません。

見るのは，毎日のニュース番組，ボクシングなどのスポーツ中継，テレビ東京系列の安上がりのバラエティ番組，一見真面目だがぶっとんだ内容の教育テレビの外国製の番組などです。あと，WOWOWの映画とか，教育テレビの英語講座とか。また，職業柄，計算機だとかマルチメディアなどに関する番組もあれば見ます。

特に有益な情報を得ようとして，計算機関連の番組を見ようとしているわけではないのですが，ときには猛暑でもひやっとすることがあります。この間も，教育テレビで3人の若手がいろいろと話していましたが，ハッカーの話題になったときに，司会役の人が，「人の名前でログインして計算機を使うぐらい可愛いもんだ，許せる」といった意味のことをいいました。

そんな発言をNHKで放映していいのでしょうか？　というのは，計算機，とりわけUNIXマシンなどの教育をするときの基本だからです。ユーザーIDがどういうもので，パスワードを他人に知られるとどう問題で，ネットワークも現実社会と同じでさまざまな法律が適用されうる世界で……。

問題は，この司会者が，そのようなまっとうな教育を受けていないらしいということで，その結果，ネットワーク社会の現状や問題点を認識していないというところにあるのです。さらに奇っ怪なことは，それを聞いた専門家らしき出席者が，「いやー，それはまあ気持ちはわかりますが，ちょっとー」とか「それをいっては，まずいですよ」なんてことさえいわず，どうも理解して相槌を打ってるふうなのです。あっけにとられました。

「あれは駄目，これは駄目」式

人になにかを教えることは難しいものです。そして，教えるべきマナーや道徳とかルールなどは，計算機やネットワーク関連でもたくさんあります。

パソコン通信や計算機ネットワークでは，不特定多数の人との交流が可能で，さまざまなトラブルが発生しがちです。そこで，さまざまなネチケット(ネットワークのエチケットのことだそうです)があります。

いちばん手っとり早いが，あまり望ましくない教え方があります。それは，あれはしちゃ駄目，これはこうするべきだ，これはこうしろ，というふうに，さまざまな禁止事項や制限事項をざっとリストアップして，そのまま覚えさせるというものです。

なぜ望ましくないかといえば，いろいろな制限や禁止事項でも，なぜそのように規定されているか，その原因にいくつかのレベルがあり，また，その結論との関係にもさまざまなレベルがありうるからです。したがって，結論としての事項を丸暗記しても，状況の変化についていけません。

たとえば，たまたま技術的な問題がいまは解決できていないからとか，利用者の間で伝統的に作られてきた習慣に反するからとか，いろいろな理由があるわけです。その理由を知らずにただこれをしたら駄目だと覚えているのでは，状況が変わったときに対応できないでしょう。

重要なのは，事実，情報，さらに事実間の因果関係をきっちり教えることです。そして，それ以上の価値判断に関わることは，各自にゆだねればいいということです。

たとえば，インターネットのニュースシステム(パソコン通信ではフォーラムとか電子掲示板みたいなものでしょうか)があります。これについて教える場合，ニュースの場というのが，不特定多数の人が見る公の場であり，親友と気軽に無責任に話したりメールをやりとりするのとは違い，さまざまな法律が適用される可能性があるということを教えることが重要だと思います。関係が深そうなところは，名誉毀損，わいせつ罪，著作権侵害などです。

ただし，教える相手が成人ならば，関連する法律をすべて列挙したり，各法律の内容を説明する必要まではないでしょう。そんなことは，社会生活を送っている人にとっては当然の義務であるからです。

さらに，技術的なことにもからめて，ユーザの行為の影響についても教える必要があるでしょう。たとえば，投稿した記事がどのように配送されて，どこの資源をどのようにして，占有していくのかということです。こういう事実まで知らなければ，情報量のほとんど含まれていない投稿をすることが，どのように問題であるかを本当に知ったことにはならないでしょう。

「朝まで生テレビ」での宮台氏

そう頻繁に見るわけではありませんが，たまに興味深いテーマがあると見てしまう

のが，テレビ朝日系列の「朝まで生テレビ」です。出席者の番組進行に対する意図とか，狙いとかが見え見えの場合でも，思わず熱くなって見てしまう性格なので……。

先日も100回記念(たったの100回？)だとかいって，相変わらずの調子でやっていました。でも，やはり，朝まで無駄テレビというか，また全部見てしまいました。

その記念番組の次の日，つまり僕の大切な日曜日を半分くらい無駄にさせてしまった出演者，それは宮台真司なる若き研究者(ブルセラ問題の解釈に関して朝日新聞でやりあっていたのをご存じの方もいらっしゃることでしょう)です。

てぐすね引いて待ち受ける手練手管に長ずる魑魅魍魎の常連たちをばっさばっさと彼は鮮やかに切り捨てました。いや，少なくとも僕の目にはそう映りました。多くの視聴者がそう見たかどうかは自信がありません。そういうふうに見えたのは，もしかしたら僕だけかもしれません。

まず，理屈っぽいことにもたえられる人でないと駄目かもしれません。実際，スタジオに来ていた一般参加者のひとりは彼の理屈についていけずに，宮台氏に「そんな議論ばかりしていないで，もっと実際に勉強しろ，なにか行動してみろ」と非難しました。ちなみに，彼はこの指摘に対して，「しゃべっているだけではない。たとえば実際に立案するグループに加わったりして行動している。勉強すべきなのはあなたです」といっていました。

で，番組における彼の主張は，まず，「ああすべきだ，こうすべきだ」などということをいい合うだけでは議論の意味がないということです。たとえば，売春する若者に対して，このような「べき論」は有効ではないということを，実際のフィールドワークの経験をふまえて主張しました。こういう行為をすると将来このようになるケースが多いとか，さまざまな情報を提供して，それをもとにして，本人に道を選ばせるのがよいといいます。そして，議論すべきなのは，そのような自主的な選択が可能なシステムをどのようにして構築するかということだと主張します。

これは，まったくのところ，至極もっともな主張であると思えます。単に，結論の部分をとってきて，ああするのはよくない，こうするのはよくないといっても，教えられるほうで，前提となる事実や，関連する情報を知らなければ，(先ほどのNHKの番組の出席者ではないですが) この程度は可愛いから許せるなどとむしろおかしな判断をしてしまうのではないでしょうか。



illustration：Haruhisa Yamada

先に，なにをしたら駄目，こうしたら駄目，というふうに明示的に羅列するのはまずいということの理由を，状況が変化したらついていけないからと書きましたが，教えられる側の心理的な面も考える必要があるようです。

つまり，それ以前に，価値判断も一緒くたにしたそのような「教え」というものは，全体主義的国家ではない限り，あるいは，家庭内で親が絶対的な権力で子供に指導する場でない限り，そもそもそれほどの効力を期待できない手法なのかもしれないということなのです。

ゲリラ的キャンプでの眠り

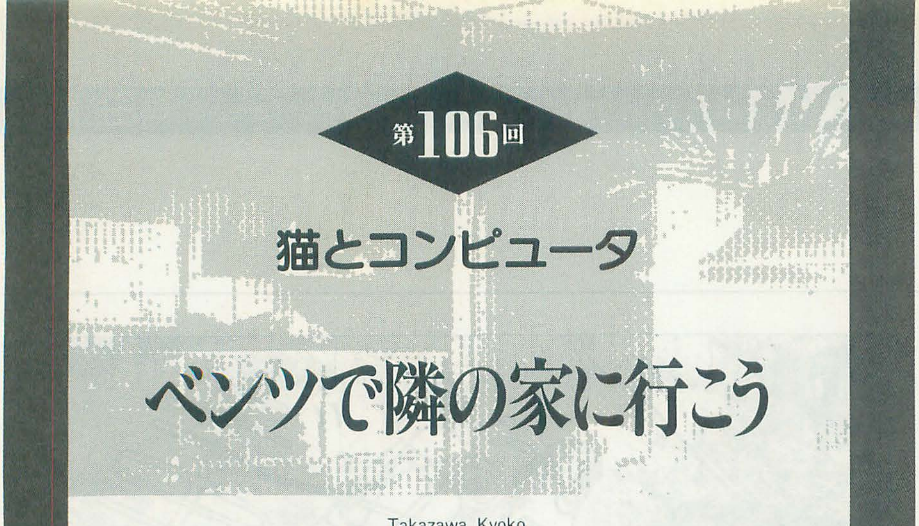
「明示的にルールを与えず，さまざまな事実とその間の関係を与えて知的な行動が生まれるのを期待する」というと，これは完全に例の話になります。例の話というのは，人工知能的な手法を批判して，人工生命的な手法をアピールするときにいわれる主張のことです。

ひと言でいえば，創発ということですね(創発については今年の5月号で述べているのでここでは詳しく触れません)。主体性を重んじるといってしまえば，ひと言で終わりですが，情報を十分に与えて，湧き起こってくるものを重視する。一方で，湧き起こるものをきっちりと受け止めるシステムを構築する。これは，相手が動物であろうが，人間であろうが知能機械であろうが，重要なことに思えます。

計算機に対しても，ルールを与えないでいこうという流れがあるくらいですから，人間に対する場合はいうまでもないのだという浅はかな結論(むろんそれだけでは世の中うまくいくはずがないのですが)に満足して，やっとの思いでみつけた場所に張ったテントの中でぐっすりと眠りにつくのでした，翌朝5時半，海岸の所有者(？)にみつかり，1000円払わされるまで。

しかも，キャンプ場にテレビをもち込むべきでないというような「べき論」の無効性をテレビがいうこと自体の意味に関してまで思いは至らずに……。

e-mailアドレス
ari@info.human.nagoya-u.ac.jp
NIFTY-ServeやPC VANから送信するときは，前者はINET:を後者はINET#を上記のアドレスの前につける。



第106回

猫とコンピュータ

ベンツで隣の家に行こう

Takazawa Kyoko
高沢 恭子

新しい機器が登場する度に生活は便利になっていきます。ただ，生活が快適になるかは別の問題です。皆がどのようにその機器を扱うかでずいぶん変わってきます。それが，パソコンの世界では……。

昼下がりの移動電話

6月，名古屋から東京に向かう新幹線は空席がいくつもあった。この時期は鉄道会社も利用状況から閑散期と呼ぶそうで，ほとんどがビジネス客と思われた。

私の席は禁煙車両のなかほどだった。乗車してすぐに，私の斜めうしろの窓がわにすわっていた男性のもとに，電話がかかってきた。携帯電話である。

彼はその場で電話を受け，通話をすませた。するとまたすぐにつぎのベルが鳴って話をはじめた。それを終えると，こんどは自分からどこかへ電話をかけてあれこれ打ち合わせをしている。すこし間をおいて，またベルが鳴る。

どうやら進行中のしごとについての連絡が刻々と入ってくるらしく，新大阪からずっとこの調子だったようだ。

新幹線に乗ると，車内放送での案内やお願いのなかに，携帯電話はデッキで利用してほしいというひとことがある。たぶんこの人だっていつも耳にしているだろう。知っているけれどチョッとだけという違反は私にもある。

彼は受けた連絡によって変わってきた対応を，さらに別の人に伝えている。

それにしても，まわりの人たちは気にならないのだろうかとあたりを見まわしたところ，初夏の午後2時，乗客のほとんどの人たちが眠りこんでいた。

これでは車内もデッキも変わらないことになるし，彼もそれを承知で移動事務所よろしく電話を使いつづけたのだろう。

歩行電話で実況中継

携帯電話を，わが家も誰かに試用してもらおうかと思うことがある。

よく活動する人ほど動きまわるから，コードにつながれた電話機だけをたよりにしていたのでは，連絡がどうしても遅れがちになる。いつでもどこでも，すばやく，しかも本人とだけ通話できる携帯電話は，その点さぞかし便利だろう。

でも少しためらいがある。それを使っている人に出会うと，ちょっと違和感を持つからだ。なぜなのだろう。

スーパーでTシャツを選んでいたお母さんが，帰宅したらしいこどもに電話で指示をあたえていた。アキバで若い人が，歩きながら，受話器ごしに目的の店までの道案内をしてもらっていた。

人気も上々のニューアイテムだから，愛用する人がふえているのに，少し気になる点は何か。

新幹線の車内放送では，「ほかのお客さまの迷惑になりますので」デッキで使用してくださいといっている。何が「迷惑」なのかを考えたがハッキリわからない。電話の話し声がウルサイのか。周囲と無関係なヒトリ話が，なんとなく無作法なのか。それなら，乗客同士の声高な会話も根本的に変わるものではない。

都心の横断歩道で，携帯電話片手にしゃべりながらやってくる男性とすれちがうこともある。少しヘンなのは，やはり，周囲との調和だろうか。そういうことをするにはふさわしい場所があるといった，公衆の場でのエチケットに反するような感じがあるからだろうか。

ファーストフード店で立食する風景や，自販機の前で飲料を飲むのも，はじめは見なれないことだった。だが少しずつ日常にとけこみ，お互いの抵抗感もなくなるころには，それなりのエチケットもできた。手軽で便利，ちょっとした解放感をいまは多くの人が受け入れている。

携帯電話もいつかそうなるだろうか。

ひとり1台の時代

ヘッドホンをつけて外出する人に，なかなかなじめない時期もあった。

とくに電車に乗り合わせたときなど，耳をふさいだスタイルが，私は社会的な交流を拒みますといった感じにとれた。その上，外までもれるほどの音で音楽を聞いている人などには，周囲で危険なことが起きてもすぐわかるのだろうかと，よけいな心配をしたものだ。

それもだんだんファッションとして見なれ，読書やイネムリと変わらない車内の景色のひとつと思えるようになった。

ヘッドホンで何を聞こうと大きなお世話だ。音楽でなく英会話の勉強をしている人も多いだろう。人との交流を拒絶しているというより，ムダな摩擦やストレスを少なくするための一案かもしれない。

携帯電話はこれとはちがう。通信できる機械だ。通信は輸送と並ぶ力がある。自分専用の通信機器があるのは，家や事務所を持つのと変わらない点もある。

パーソナルな輸送といえば自家用車。クルマと呼んで親しむ。クルマもいまはひとり1台という家も少なくない。

日本はクルマをつくる技術もすばらしく，運転者も法規をよく守るし，総合評価も合格点と思っていた。ところが日本自動車連盟の専門家が，日本のモータリゼーションはアメリカより50年遅れていると指摘していた。

ほんとうの意味での自動車を用いた生活のありかた全般を見たとき，私たちが身につけなければならない習慣や知識が，まだ

まだ足りないという。日本には機械としてのクルマの正しいあつかいかたを知らない人があまりに多いのだそうだ。

多くの人が自家用車を利用するようになって40年くらいだろうか。豊かさの象徴のように，車種もデザインも色とりどりにあふれる日本のクルマ。じゅうぶん普及し活用されていると思っていたのに，理解もあつかいも劣っているというのは，日本ならではの特徴があるからではないか。

みんないっしょが悪い

三重県は正しくは東海地方だそうだが，ドライブでおとずれる近畿一円の道路が滞ることはまずない。すみやかで快適，風景の美しさがいっそう目にしみる。

アメリカの国土や道路の広さとは比較にならないとしても，こんな悠々とした条件でいつもクルマを使えたなら，日本の自動車文明もきっと変わっていただろう。

東京の道路では，クルマは走るというより徐行と蛇行で進んでいる。また，私たちの住まいの周囲だけを見ても，夜となく昼となく，路上駐車の列で道路の片側半分はふさがれたままである。

都内の通勤にクルマを使うときは，よほど時間のゆとりがなくてはダメだ。多くの人が，ダイヤの正確さでは世界一と定評のある電車や地下鉄を使う。

休日には，ウィークデーに休ませてあったクルマがいっせいに動き出す。都心のデパートは駐車するのに買物以上の時間がかかり，郊外もレジャーの隊列ができる。

お盆とお正月をはじめ，もともとみんなが歩調を合わせる生活習慣なので，知っていながら同じパターンでクルマを使う。

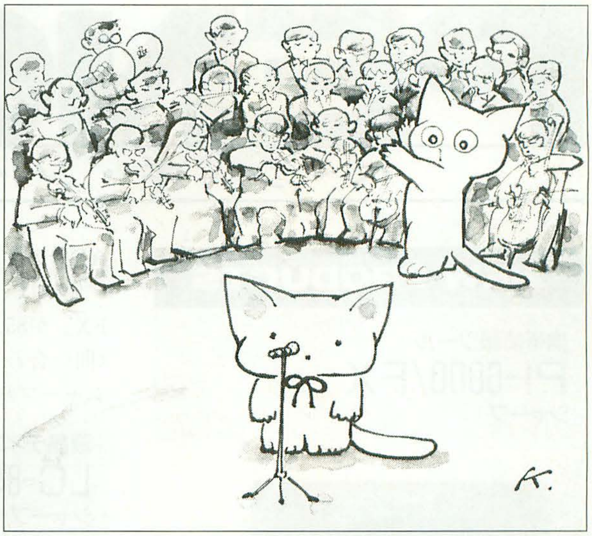
慢性的な混雑のなかでの，みんないっしょの自動車ライフ。運転技術も隣のクルマのマネをしていればいい。クルマの正しいあつかいかたを知ろうという意欲より，渋滞のきりぬけかたや，よりよいルートの選びかたがたいせつだ。

ゆとりのない道路，車庫もない人が多い日本では，クルマは見た目のカッコよさと，追い越しだけで勝負する。

パソコンが強い

ひととおり普及の終わったパーソナル輸送のクルマより，みんなが興味を向けていくのはパーソナルコンピュータであり，通信の機械である。携帯電話の人気もその流れのなかにあるらしいから，批判を早まってはマズイようだ。

新聞記事に，日本電子工業振興協会の調査による，ことし4月から6月までのパソコン出荷台数の集計発表があった。国内では108万4千台で昨年同期の52.5％増という好調のため，95年度の出荷台数の予想を430万台から500万台に修正したそうだ。


illustration : Kyoko Takazawa

テレビでもことし上半期だけで 220万台の売り上げがあったと報道していた。好成績のおもな理由は，CD-ROM によって音楽や映像，ゲームなどが豊富に楽しめるといった，マルチメディアタイプのマシンが続々とあらわれたためだ。

パソコン通信業界ではいま大きな話題がある。マイクロソフト社の新しいOS「Windows95」の脅威である。

この基本ソフトには，同社が新しくはじめる通信サービス「マイクロソフトネットワーク(MSN)」に接続できる機能が組み込まれているため，マウスクリックひとつでアクセスできる。

このことから予測される膨大な利用状況が，独占禁止法にあたるのではないかと，アメリカの司法省が3カ月間調査をつづけてきた。けっきょく，販売後のようすを見るという結論になり，1995年8月24日に予定どおりの発売がきまった。同時にMSNの通信サービスも発足する。

アメリカ司法省の独占禁止法に関する判断には，「規制することが市場発展の原動力を妨げることになってはいけない」という経済学者や法曹関係者の見解も参考にされたようだ。

「Windows95」は世界じゅうで爆発的に売れることが予測され，日本でも年内に発売される。多くの機種に搭載される予定なので，パソコンの売り上げはますますのびるだろうという。

昨夏から1年，DOS/V マシンのワガママで不可解な態度に，わが家は手をやいてきた。各部ハードが自由に改造できるなんて文句にノセられて，クルマほどのお金をかけながら，あっちでぶつかり，こっちでエンコし，修理，入院もいくたびか。まったくクルマ並みだ。

退院してくるとケロッと元気になるけれど，若いくせに足腰が弱いらしく長つづきしない。すこしの重みで急にまた，「立ち上がれません，調べてください」なんていう。「そんなことないだろう，どこが悪いっていうの」とアタマや背中をのぞいてみると，自分勝手に屈伸運動をしたアトがあり，前とはようすがちがっている。

「しょうがないなあ，もういっぺんやってごらん」とハッパをかけると，「あ，できました」なんて別人みたいになる。

いま人気の，メーカーがはじめからセットして組み立てたタイプのWindowsマシンは，もうちょっと使いやすいそうだ。

それでも，別の新しいソフトを組み込もうとするときは，なかなかめんどうなことも多いとみえて，狛江のアニキも奮闘しているようだ。

「なんだか，隣の家に行くのにベンツのエンジンかけてるみたいなんだよネ」というのがアニキの感想だ。

DOS/Vマシンを使うとき，これほど大ゲサでなくてもいいのにと思うことがよくある。散歩のために登山装備をする必要はないし，アニキではないが，隣の家ならサンダルばきでじゅうぶんだ。

とはいえトラブル多発は運転未熟によるもの。早く名ドライバーになって，隣の家にもクルマで行こう。

NEW PRODUCTS

携帯情報ツール
PI-6000/FX
シャープ



PI-6000FX

シャープは携帯情報ツール“アクセスザウルス”「PI-6000」「PI-6000FX」を発売した。

「PI-6000」の入力はペンで行い，手書き認識も楷書体以外での認識も可能になった。学習登録機能は，くせ字などで判別しにくい文字も最大100文字まで登録できる。質問に答えていくだけで和文88種類，英文89種類の各種レターが作成できる機能を搭載。また，パソコンの表計算ソフト「MS-Excel ver.5.0」で作成したデータを光通信インタフェイスや専用通信ケーブルを通じて本機に転送し，参照できる。通信機能としてはザウルスネットを搭載し，特にNIFTY-ServeへのアクセスはGUIでの操作が可能。また，FAX送信ソフトも内蔵しており，別売りのFAXモデムを使ってのFAX送信もできる。ほかにも，国語，漢和，英和，和英の4種類の辞典を内蔵したり，パソコンから専用プログラムを本体に読み込んで実行するAdd in機能も搭載されている。メモリ容量は1Mバイトでユーザーエリアが約717Kバイト。ICカードスロットを1基装備しており，メモリの増設なども可能。

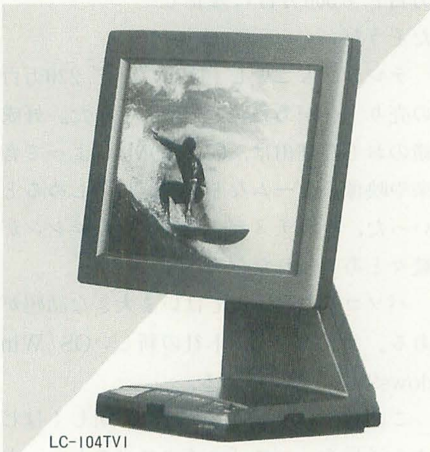
「PI-6000FX」は「PI-6000」にFAXモデムを追加したもの。

大きさは147mm(幅)×87mm(奥行)×17mm(厚さ)で従来機「PI-5000」に比べ約83%小型化し，重さが約195gと約78%の軽量化を実現し，さらに携帯性を高めた(ただし，FAXモデム部分は除く)。

価格は「PI-6000」が69,000円，「PI-6000FX」が85,000円(それぞれ税別)。

〈問い合わせ先〉
シャープ㈱　☎06(621)1221, 03(5261)7271

液晶ディスプレイテレビ
LC-84TV1/LC-104TV1
シャープ


LC-104TV1

シャープは液晶ディスプレイテレビ「LC-84TV1」「LC-104TV1」を発売した。

両機とも画素数921,600ドットのTFT液晶パネルを採用し，液晶パネルの透過率向上と高輝度長寿命バックライトの採用で最高画面輝度300cd/m²を実現した。音声出力は1Wの5cm丸型スピーカー1個で行う。接続端子はDC入力端子，FMアンテナ端子，映像入力端子などを装備している。また，テレビ以外にもFM/AMラジオも楽しめる。ほかにも，セットした時刻に電源をつけることができる生活プログラムタイマ機能やちょっとした時間管理に便利なお知らせチャイム機能，オフタイム機能などが用意されている。

重さは「LC-84TV1」が約1.9kg，「LC-104TV1」が約2.2kgとかなり軽く，ディスプレイ部を動かせる回転チルトスタンドを採用することで，左右90度回転，前後角度，高さの調整が可能。

価格は「LC-84TV1」が110,000円，「LC-104TV1」が150,000円(それぞれ税別)。

〈問い合わせ先〉
シャープ㈱　0120(078)178

3.5インチ光磁気ディスク
MA-M640/MA-M540
日立マクセル



日立マクセルは次世代の3.5インチ磁気ディスク「MA-M640」「MA-M540」のサンプル出荷を始めた。

「MA-M640」はディスク容量が640Mバイト(2,048バイト/セクタ)で「MA-M540」はディスク容量が540Mバイト(512バイト/セクタ)となっている。どちらも現行の光磁気ディスク同様ISO規格に準拠しているため，下位互換機能のあるドライブでは128～640Mバイトのディスクが使用できる。現行のメディアから記憶容量が増えた理由は記録方式がマークポジションからマークエッジに変更されたこと，トラックピッチやビットピッチが狭まったこと，記録領域が広がったことにある。

発売は今秋の予定。

〈問い合わせ先〉
日立マクセル㈱　☎03(5467)9334

磁気メディア
Zipドライブ/ディスク
キヤノンソフトウェア

キヤノンソフトウェアは米アイオメガ社のZipドライブ/ディスクを販売する。

主な仕様は以下のとおり。

Zipドライブ
- インタフェイス：SCSI/パラレルポート
- ディスク回転数：2,980rpm
- 平均シーク速度：29ms
- データ転送速度：最大1.25Mバイト/秒

Zipディスク
- 記憶容量：100Mバイト
- 記憶方式：磁界変調磁気記録



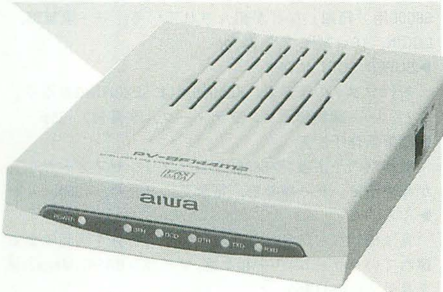
Zipドライブ

価格はZipドライブが29,800円，Zipディスクが2,680円(それぞれ税別)。

〈問い合わせ先〉

キヤノンソフトウェア㈱　☎03(3453)7965

FAXモデム
PV-BF144M2WW/WA
アイワ



PV-BF144M2

アイワはFAXモデム「PV-BF144M2WW」「PV-BF144M2WA」を発売した。

両機はデータ通信の最高速度が14,400bps，データ圧縮機能はMNPクラス5，コマンドはHayes ATコマンドに準拠している。FAX通信は最高速度が14,400bps，コマンドはClass 1/2に準拠し，G3FAXに対応している。また，通信制御には携帯電話などの移動体通信と通信ができるMNPクラス10を採用している。

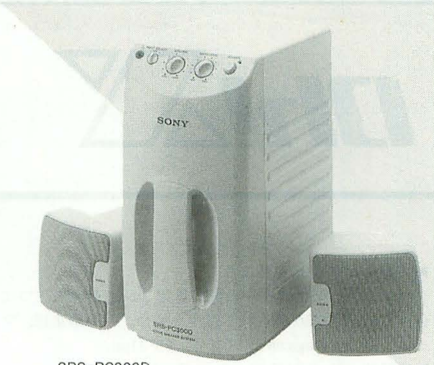
「PV-BF144M2WW」はPC-98シリーズ，DOS/V機用，「PV-BF144M2WA」がMacintosh用で通信ソフトなどが付属している。

価格はどちらも19,000円(税別)。

〈問い合わせ先〉

アイワ㈱　☎03(3371)7981

アクティブスピーカーシステム
SRS-PC300D/SRS-T50PC
ソニー



SRS-PC300D



SRS-T50PC

ソニーはアクティブスピーカーシステム「SRS-PC300D」「SRS-T50PC」を発売した。

「SRS-PC300D」はウーファとサテライトスピーカー2個で構成される。ウーファは出力が15Wで，再生周波数は50～200Hz。また，重低音を調節できるバスブーストとコントロール機能を搭載し，入力端子として2系統3入力，出力端子として2系統を装備している。サテライトスピーカーは1個が5Wの出力が可能で，再生周波数は200～20,000Hz。ウーファとサテライトスピーカーともに防磁設計がなされている。

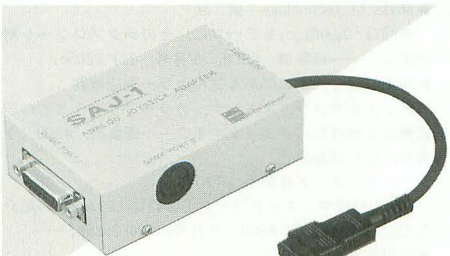
「SRS-T50PC」は本体左右にスピーカーを内蔵し，リフレクタにより1ボックスでステレオサウンドが楽しめる。出力は4Wで，入力端子が1系統，再生周波数は150～20,000Hz。電源はACアダプタも付属しているが，単3電池4本での駆動も可能。

価格は「SRS-PC300D」が20,000円，「SRS-T50PC」が8,000円(それぞれ税別)。

〈問い合わせ先〉

ソニー㈱　☎03(5448)3311

アナログジョイスティックアダプタ
SAJ-1
システムサコム



SAJ-1

システムサコムはアナログジョイスティックアダプタ「SAJ-1」を発売する。

同機はAT互換機用ジョイスティックをアタリ準拠のポート出力に変換する。データの転送方法は電波新聞社製の「XE-1AP」と互換性がある。特殊な機能として，ボタンやXY方向の入れ替えができたり，XY方向のアナログ値の補正ができたり，AT互換機のキーボード出力(一部)をボタン出力に変換できる。また，ジョイスティックはFLIGHT CONTROL SYSTEM(スラストマスター社製)，Wing Man EXTREME(ロジテック社製)，フットペダルはRUDDER CONTROL SYSTEM(スラストマスター社製)などで動作が確認されている。

価格は15,000円(税別)。

〈問い合わせ先〉

㈱システムサコム　☎03(3797)0211

液晶プロジェクションテレビ
XV-R36/XV-R43
シャープ



XV-R43

シャープは液晶プロジェクションテレビ「XV-R36」「XV-R43」を発売する。

「XV-R36」が36型，「XV-R43」が43型のワイドテレビで，両機ともに同社の14型ブラウン管テレビ(36.4cm)とほぼ同じ奥行き(36型：35.0cm，43型：38.5cm)，重さも同社の28型ブラウン管テレビ(39kg)とほぼ同じ重さ(36型：35kg，43型：41kg)を実現した。液晶パネルは，614,440ドットの4型のTFT液晶パネルを採用している。機能面では，テレビを見ながら文字放送の番組を呼び出せる新ニュース機能，4：3の映像などを最適なワイド画像に切り替える“瞬速”オートワイド機能が用意され，水平高画質回路によりワイドクリアビジョン放送にも対応する。音声出力は15Wの12cm丸型スピーカー2個で行う。接続端子はビデオ入力3系統，S映像入力端子2系統，BS専用入/出力端子などを装備している。

価格は「XV-R36」が348,000円，「XV-R43」が398,000円(それぞれ税別)。

〈問い合わせ先〉

シャープ㈱　0120(078)178

FILES Oh!X

このインデックスは，タイトル，注記――著者名，誌名，月号，ページで構成されています。秋の夜長にたまにはゆっくり読書でもいかが。

一般

▶NEWS
「NetWorld＋Interop95 Tokyo」のレポートや電話料金定額制の話題など，最新のトピックを紹介。――編集部，ASAHIパソコン，8・15/9・1号，8-11pp.

▶NEWS & VIEWS
アメリカで第5位のパソコンメーカーで，日本法人を設立した「GATEWAY2000」の紹介。――編集部，ASAHIパソコン，8・15/9・1号，14-15pp.

▶特集1　姿を現したWindows95の正体
ようやく発売されるWindows95をベータ版で最終チェックし，発売後の変動が予想される周辺状況を探る。――山田祥平ほか，ASAHIパソコン，8・15/9・1号，18-29pp.

▶98ユーザーのためのマッキントッシュ教室　19
今回はPC-98とMacintoshのバスやインタフェイスなど規格の違いについて解説する。――荻窪圭，ASAHIパソコン，8・15/9・1号，36-39pp.

▶特集2　ジャンル別楽しいCD-ROMソフト大全
CD-ROMをエデュテインメント，教養，ゲームなどのジャンルに分け，それぞれのお勧めソフトを紹介。――森英二ほか，ASAHIパソコン，8・15/9・1号，119-131pp.

▶特集3　台湾 vs. シンガポール
台湾とシンガポールのパソコン界の現状とこれからどうなるかをレポートする。――編集部，ASAHIパソコン，8・15/9・1号，132-137pp.

▶ハードウェアFLASH！
各社の最新パソコンやディスプレイなどの周辺機器の新製品を紹介。――編集部，LOGIN，16・17号，44-47pp.

▶THE NEWS FILE
各種イベントのレポートや，プラズマトロンディスプレイ，ゴーグル型ディスプレイ発表のトピックなど。――編集部，LOGIN，16・17号，48-53pp.

▶特集　ハード・オブ・ダークネス
PCの組み立て完全マニュアル，一体型パソコンの改造，エミュレータの紹介の3部に分けてハード分野にアプローチする。――編集部，LOGIN，16・17号，171-201pp.

▶インターネットの心
インターネットにアクセスできるカフェの紹介や，用語解説，接続ガイド，ホームページ紹介など。――編集部，LOGIN，16・17号，226-229pp.

▶台湾に世界市場を見た!!
「消費性育楽多媒體展」というゲームショーのレポートから台湾のゲーム市場を探る。――編集部，LOGIN，16・17号，234-237pp.

▶くねくね科学探検隊　第25回
今回は金の採掘の歴史と現状を紹介する。――鹿野司，LOGIN，16・17号，248-251pp.

▶特集　厳選！　最新パソコン周辺機器ガイド
周辺機器の基礎知識の簡単な解説と周辺機器カタログ。――編集部，コンプティーク，9月号，21-31pp.

▶こだわりゲーム年代記
今回はパソコンゲームに登場する探偵からパソコンゲームの歴史的背景や魅力を探る。――与志田拓実，コンプティーク，9月号，100-101pp.

▶NEWS COLLECTORS
任天堂のウルトラ64のコントローラを予測したり，バーチャルボーイの中身を分析したり，業界の最新情報を紹介。――編集部，電撃王，9月号，28-33pp.

▶特集　美少女ナンパ・ゲーム大ヒットの秘密
美少女ナンパゲームに関する考察と「ときめきメモリアル」と「同級生2」の攻略法。――編集部，電撃王，9月号，34-49pp.

▶特集　ビデオCDを見よう！
パソコン，新世代ゲーム機，専用プレイヤーの3つに分けてビデオCDが見られる環境を紹介。――編集部，マイコンBASIC Magazine，9月号，25-31pp.

▶超記憶メディアMDデータ vs Zipディスク
MDデータとZipディスクの特徴を紹介し，比較する。――編集部，マイコンBASIC Magazine，9月号，40-43pp.

▶コンピュータ・ミュージック・ショー・ケース
IBMのAptiva Music Visionやカシオ計算機のGM音源を紹介。――編集部，マイコンBASIC Magazine，9月号，62-63pp.

▶パソコンレクチャーX
PD，MO，HS，MD，Zipなどの大容量メディアについて漫画で解説する。――くりひろし，マイコンBASIC Magazine，9月号，64-67pp.

▶ナムコ・ワンダーエッグ＆たまご帝国大解析
ナムコのテーマパーク，ワンダーエッグとたまご帝国のアトラクションを紹介する。――編集部，マイコンBASIC Magazine，9月号，147-151pp.

▶特集　インターネット＆パソ通
インターネットに関するQ＆A，ホームページ紹介，キーマンへのインタビュー，業界トピックスなどからパソコン通信やインターネットの現状を紹介する。――竹本隆ほか，I/O，9月号，17-40pp.

▶ポインティングデバイスの選び方
トラックボールやポイントパッド，タッチポインタ，可変マウスなどのポインティングデバイスを紹介。――清水亮，I/O，9月号，53-56pp.

▶CADの基礎知識
用途別CADの一般的な解説とCADソフトとプロッタの紹介。――森羅万象，I/O，9月号，57-71pp.

▶マルチメディアの音質アップ
各社のアンプ内蔵スピーカを，音楽CDを再生したり，FM音源を再生したりしてテストする。――服部裕子，I/O，9月号，104-109pp.

▶DeskTopMusic入門　6
今回はDTMで演奏を聴いて楽しむ方法を紹介する。――あまだたかし，I/O，9月号，124-127pp.

▶MultiMedia Watching 21
インタラクティブテレビのシミュレーション実験や，プラズマトロンディスプレイ開発の話題などを紹介する。――奥野雅之，I/O，9月号，129-131pp.

▶特集I　8・24Windows95が起きる！
Windows95英語版のレポートやWindows95の登場によって変化する環境のチェック，対応ソフトの予測レビューなど。――編集部，ASCII，9月号，189-212pp.

▶Abyss of Technology　第1回
今回は「OpenDoc」をテーマに，そのテクノロジーを解説する。――編集部，ASCII，9月号，221-226pp.

▶特集II　今，そこにあるフルカラー周辺機器
デジタルカメラ，スキャナ，カラープリンタなどの周辺機器を機種別にチェックする。――編集部，ASCII，9月号，227-249pp.

▶インターネット膝栗毛　ROUTE 6
FTPの解説や，ネットワーク上の雑誌「E-Zine」の紹介など。――編集部，ASCII，9月号，294-298pp.

▶Digital Beat Zoo
今回はヤマハのFM/GS音源＋DSPを搭載した拡張カード「Sound Edge」とXG音源対応ドータボード「DB50XG」のレビュー。――編集部，ASCII，9月号，313-316pp.

▶脳型コンピュータを作る　第2回
今回は前回に続いて，脳をコンピュータ的に説明するとどんなものになるかを解説する。――編集部，ASCII，9月号，317-324pp.

▶帰ってきたバカパパ BUPPiN
パソコンを使うときに体が楽になるグッズを紹介する。――編集部，ASCII，9月号，362-363pp.

X1/turbo/Z

X1turboシリーズ

▶Get Coin！
海の底に落ちているコインをブロック崩しの要領で取っていくワンキーゲーム。――HELL，マイコンBASIC Magazine，9月号，115-116pp.

X68000

▶未確認クリエイターズ
名作解剖実験団のなかで松本岳美氏が作成したX68000用「箱船」などが紹介されている。――編集部，LOGIN，16・17号，220-223pp.

▶SUPER SOFT INDEX
新作ソフトの予定表。X68000用は「EXCITINGみるく」など。――編集部，コンプティーク，9月号，125p.

▶電撃新作予定表
機種別の予定表。X68000用は「プリンセスメーカー」が発売予定。――編集部，電撃王，9月号，186p.

▶ぶんぶん
食虫植物などに食べられないように自分を画面端まで連れていくゲーム。――和々草，マイコンBASIC Magazine，9月号，116-118pp.

▶ブランディッシュ3～はぐれ狼たちの伝説～
NAGDRV 2＋GS音源用の音楽プログラム。――上古仁志，マイコンBASIC Magazine，9月号，124-127pp.

▶SUPER SOFT Hot Information
X68000用は名作必携ソフトとして「ファンタジーゾーン」を紹介し，発売予定表には「亜美ちゃんの写真集」など。――編集部，マイコンBASIC Magazine，9月号，とじ込み付録11p.

▶ONLINE SOFTWARE INDEX
大手ネットにアップロードされたプログラムを紹介する。X68000用はSCSIデバイスドライバ「susie.x」や拡張FDドライバセット「9SCSET.LZH」など。――編集部，ASCII，9月号，391-392pp.

▶SX-WINDOWプログラミング　第23回
今回はいままでの連載で登場したソースについて補足説明をしていく。――吉野智興，C MAGAZINE，9月号，126-131pp.

ポケコン

PC-E500

▶移植版 JankenWar 2′
じゃんけんと落ちものパズルを組み合わせた対戦ゲーム。――eyes，マイコンBASIC Magazine，9月号，119-120pp.

参考文献

I/O　工学社
ASAHIパソコン　朝日新聞社
ASCII　アスキー
コンプティーク　角川書店
C MAGAZINE　ソフトバンク
電撃王　主婦の友社
マイコンBASIC Magazine　電波新聞社
LOGIN　アスキー

QUESTION and ANSWER

Oh!X 質問箱

Q **DMA転送はやたら敵視されているようですが，なにか弊害があるのでしょうか。X68000はCRTCをいじっていろいろできるみたいなんですが，具体的にどうやるのでしょうか。Z-MUSICのムックはもう在庫がないらしいんですが，パソコン通信もしてませんしまわりにXユーザーもいない僕がZ-MUSICのマニュアルや標準のPCMデータを手に入れるにはどうすればいいんでしょうか。**

広島県　湊　和久

A DMAが嫌われる主な理由は10MHz機を除いて，DMAの転送速度より，CPUの転送速度のほうが速くなっているからです。DMAを大量に使ったプログラムの場合，マシンのCPU性能が上がっても速度は変わりません。

また，DMAが最高性能を発揮するのはバーストモード時ですが，この状態ではCPUはまったく停止してしまいますから，タイマ割り込みで動作している音楽ドライバなどが正常に動作しなくなります。

CPUによって使い分けるというのが正解ですが，10MHzを使用する場合でも，CPU転送とDMA転送の速度差はさほど大きなものではありません。

とはいえ，DMAというのはシステムに密着したところに使われています。FDからの読み込みやAD PCM再生などのDMA転送に依存した処理が実行できないということがX68000でメモリが簡単に増やせない要因のひとつにもなっています。そういったことの積み重ねがあまりよくない印象を与えているのでしょう。

CRTCは基本的にはデータを書き込んでいくだけなのでInsideX68000，書き込むデータについてはOh!X1994年7月号に付属のCRTC.Xを参照してください。

Z-MUSICはフリーソフトウェアセレクションvol. 2のCD-ROMに収録されていましたが，現在ではこちらを入手するのも非常に難しいものがあります。Z-MUSIC関係の書籍はコピーフリーになっていますので，なんとか友達を見つけてコピーするなりしてください。

Q **X68000のアセンブラでスプライトを表示する簡単なプログラムを作って走らせたところ，PCGエリアとSPスクロールレジスタの書き込みでバスエラーが出ます。InsideX68000でも特に注意書きがなく困っています。**

大阪府　入船　信章

A 同封されていたプログラムを実行すると確かにバスエラーが発生します。

こういう症状の場合，普通はスーパーバイザモードへの移行を失敗しているとかいうのが相場なのですが，そのあたりはちゃんとしてますし，別にアドレスは間違っていませんし……。はて？

さて，原因ですが，このプログラムには画面モードが設定されていませんでした。これを通常の768×512モードから立ち上げるとバスエラーが発生してしまいます。スプライト関係のI/Oは画面横ドット数が512を超えるとメモリ空間から消滅してしまいますので注意が必要です。

これはテストプログラムだからこそ起きたバグといえるかもしれません。

Q **CISCとRISCは具体的にどう違うのですか。「命令が少なく，速い」ことで機械語を操作できないのはどういう理由からですか。**

山形県　金渕　満

A RISCでも機械語は操作できます。ただ，効率が上げにくいというだけの話です。

それと，構造的にRISCとCISCに特に違いなどはありません。RISCの命令が少ないというのも嘘です。むしろ「アドレッシングモードが少ない」のほうが適切です。

機種によってはマイクロプログラムを思わせるコードが多い場合があること，機種によっては，CPUの内部構造を熟知していないと効率のいいコードが書けないこと，最適化の非常に面倒なことまで把握しておく必要があることなどで，コンパイラにはかなわないことが多いということになっています。

最近のRISCは内部も綺麗に作られており，基本命令の機能なども68000などと比べても多く，機能的に遜色はありません。むしろ一部のCISCよりは機械語プログラムを作るのは遥かに簡単かもしれません。理屈では上級者なら機械語を使うほうが実行時の効率はよいはずです。メモリアクセス量を減らせば性能は格段に上がります。

しかし機械語で十分な性能を発揮させるにはパイプラインの管理が大変とか，レジスタの使いこなしに経験が必要といった，難点があります。それらを十分踏まえておかないと，コンパイラの出力したコードにも劣るものしかできないでしょう。

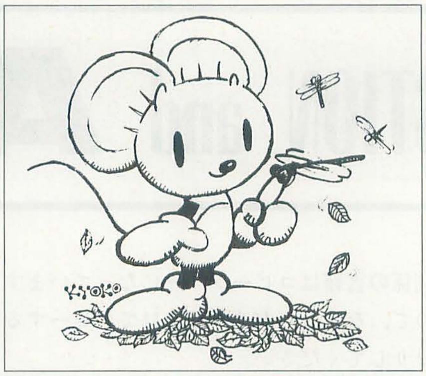
むしろ，コンパイラが異様にがんばっているので，それより圧倒的によいコードを書くのは困難です。むしろ開発コストが見あわないので機械語はほとんど使われないだけでしょう。　　　　（中野　修一）

質問にお答えします

日ごろ疑問に思っていること，どんなことでも結構です。どんどんお便りください。難問，奇問，編集室が総力を挙げてお答えいたします。ただし，お寄せいただいているものの中には，マニュアルを読めばすぐに解答が得られるようなものも多々あります。最低限，マニュアルは熟読しておきましょう。質問はなるべく具体的に機種名，システム構成，必要なら図も入れてこと細かに書いてください。また，返信用切手同封の質問をよく受けますが，原則として，質問には本誌上でお答えすることになっていますのでご了承ください。なお，質問の内容について，直接問い合わせることもありますので電話番号も明記してください。

宛先：〒103　東京都中央区日本橋浜町
3-42-3
ソフトバンク株式会社出版部
Oh!X編集部「Oh!X質問箱」係

STUDIO X ON AIR



FROM READERS TO THE EDITOR

創作活動に力が入る季節がやってきました。過ごしやすくなっていままで以上にプログラミングがはかどるかな。食欲旺盛な季節でもありますので，冬の寒さに備えて皮下脂肪を蓄えるのもいいかもしれませんね。

◆CD-ROMドライブを購入しました。これでとりあえず，雑誌付録のCD-ROMの中身が見れます。8月号の「CPKPLAY.X」と「CDG-TX4」の紹介記事を読んで買ってしまった感じです。ちなみに購入したCD-ROMドライブは，ロジテック「LCD-440」で，中身は「CDG-TX4」と同じドライブのものです。　中村　正夫(26)神奈川県

◆8月号の付録ディスクに収録されていた「CPKPLAY.X」はよかった。いずれ誰か作るだろうと思っていたが，まさかOh!Xから発表されるとは……。　堀尾　忠教(22)鳥取県

◆「CPKPLAY.X」は8月号の付録ディスクに収録されていたものの中では，最高によかったです。SEGA SATURNを持っててよかったと思いました。おかげさまで「パンツ」のエンディングを拝むことができました。さらに「グランチェ」とか「サターン通信」のCDも見れましたよ。一部のCPKファイルは「未対応のフォーマットです」と怒られますが，未対応のフーォマットにも対応し，音楽と同期を合わせるように発展していってほしいと思います。作者の菊地さんがんばってくださいね。萩原　保憲(28)神奈川県

◆「CPKPLAY.X」がきっかけになって，CD-ROMドライブを買いました(笑)。購入後，さっそく実行……遅い。再生前に一括デコードするようにすると速くなるんでしょうかね。HANIM.X(DōGA)って速いじゃないですか。

中野　晋一(23)山口県

ちょっと怪しい「CPKPLAY.X」。結構好評で，読者の皆さんに満足していただけているようでなによりです。

◆「PICTパズル」を遊びました。最初はルールを読んでもわけがわからず戸惑いましたが，いまでは，NORMALモードの中盤戦に差し掛かっています。BGMもゲームにマッチしてよいっス。FM音源版もSC-55版と全部いいけど，特に3番目のBGMがウォーって感じです。それから「EX-System体験版」を体験してみました。放射光機能がすごいですね。製品版を使ってみたくなりました。　三沢　弘之(24)神奈川県

◆8月号の付録ディスクはとてもよかったです。「PICTパズル」は「倉庫番リベンジ」と同じくらい面白かったです(「マリオのピクロス」がいらなくなりますね)。SX-WINDOW関係のソフトはまだ使っていませんが，div.xなど使ってみたいソフトが多いので，どんどん使おうと思っています。　西山　浩(20)東京都

◆「PICTパズル」は，記事中に書かれているとおり，問題がかなりつまらないと思います。システムはかなりよいのですが。ただ，20×20マスでは，少々もの足りなくなりそうで残念です。

横堀　正敏(31)埼玉県

オリジナル問題を募集していますし，横堀さんが考えた面白いパズル問題を投稿してみては？楽しみに待っていますよ。

◆「EX-System」は，8月号の付録ディスクの体験版で大いに期待に応えてくれるものでした。特にアナログマスクを使用したパターンブラシはすばらしい効果です。10MHzではちょっと重い処理があるのが残念ですが，十分耐えられる範囲だと思います。　長崎　望(20)埼玉県

◆「EX-System」はとても面白いです。そのためにCD-ROMドライブがほしくなってしまいました。「CPKPLAY.X」は面白そうですが，私はPlayStationユーザーなので，PlayStationのアニメーションローダも作ってほしいです。そんで「鉄拳」や「クラクラ」のエンディングアニメーションをウハウハなんていいと思うけどなあ。

原　弘樹(21)東京都

専用のハードでガシガシ展開するPlayStationのアニメーション再生は，ちょっと難しいかな。

◆「div.x」の存在意義については，ちょっと疑問に感じるところがありますが，「すべてをSX-WINDOW上で」「情報量増加」という観点から考えると納得できます。特にこの「情報量」は，SX-WINDOWを使ううえでの最大のメリット(Human68kに比べて)だと思うんです。12ドットフォントも皆がきれいに使えるようになったし。そして，ここからは要望なんですが，「PICTパズル」をSX-WINDOW上に移植できませんか？　SX-WINDOW＋12ドットフォントなら巨大ロジックも遊べると思うのですが。20×20じゃつまらん！　50×50マスが最低ラインでしょう。そうすると問題作成が大変でしょうけど。

遠藤　勝博(25)宮城県

◆「div.x」は，どうみてもWINDOWSのファイルマネージャと同じだが，なにか違う。SX-WINDOWの作り方がいいのか，健人さんの作り方がいいのだろうか。センスがいいぞぉー。使いやすいぞー。　猪狩　友則(22)千葉県



◀渡辺　宏幸　埼玉県
調教というよりサギ行為に近いんじゃあ……。確かにウソはついてないけどさ。でも本格的な調教はこれからでしょう。がんばって責任取ってね。



◀千装　茂夫　埼玉県
相変わらずとぼけた千装さん。しかし、どうやら本格的なX68000 ACE-HD復活計画がもち上がっている様子です。今度こそ生き返らせてね。



◀岡村　直也　京都府
4コママンガのトリはやっぱこの人。あと、読者の皆さんリバイバルレビュー中の姉・第4コマ番外編を見逃さないように。



▲青木 一師 奈良県
月にススキとくれば団子でしょう(?)。浴衣姿の女の子はよしとして，月見の主役である団子がないのはいけませんねえ。



▲鈴木 道明 埼玉県
点棒がでかいのか，ミニチュアサイズのしょう子ちゃんなのか。ま，それはともかく水着姿のイラストも今月で見納めかな。ちょっと寂しいけど。



▲武田 正道 兵庫県
そして，最後の残暑見舞いイラスト。暑い暑いといい続けた今年の夏も終わり。来年はほどほどに暑いといいな。

◆私はLispが大好きです。C言語でLispインタプリタ＋正規表現を作り，テキスト処理などに使っています。「div.x」はC＋＋でLispインタプリタを実現しているので，とても興味をもちました。ソースリストを読んで勉強したいと思います。また，今後のLisp入門に期待しています。
広井 誠(33)新潟県

今月から始まった「lisp一夜漬け」。どうでしたか？

◆冷夏の予報に反しての連日の猛暑で少々バテ気味です。そんななか，8月号は瀧氏の新製品紹介や緊急座談会「パソコンゲームの未来はどうなる？」など妙に熱い記事が多く，「暑中見舞いPRO-68K」どころか，熱射病になりそうです。もはや怖いものはない，といったところでしょうか。今後の記事に期待しています。
松永 貴輝(24)大阪府

9月号からゲーム関係で，いくつか新しいコーナー，連載が始まっています。感想を聞かせてね。

◆8月号の緊急座談会「パソコンゲームの未来はどうなる？」では，本当に目をみはった。「ゲームを作るためにはゲーム学校に通うしかないのだろう」というのは，深刻な問題だと思う。独学の叩き上げという人が，私の友人にいるが，今後そういった人々はいなくなってしまうのだろうか。
河合 健一(24)神奈川県

いなくならないとは思いますが，割合は減るでしょうね。

◆「THE USER'S WORKS SPECIAL」はよかったです。すでに他誌で紹介されているものもありましたが，とにかく市販ソフト，新作ソフトにこだわらずにどんどん取り上げてほしいですね。
深見 満彦(18)和歌山県

なにかお勧めのソフトがあったら情報をよろしくね。

◆「CPKPLAY.X」での菊地さんのがんばり，SCSI-2ボードでの瀧さんの喜びようが伝わってくるような誌面ですね。かまたさんの論にはうなずけるものがあります。こんなOh!Xが私は好きです。さて，シャープさんは"実りの秋"を迎えさせてくれるのでしょうか？
内海 秀樹(27)大阪府

◆「アマチュアCGA現状論」を読んで考えたのですが，「あっといわせる技巧」があれば意味なんてなくてもいいと思います。「CGを使って映画を作ろう」というわけではないのですから，すべての作品にテーマを求める必要はないと思うのです。ましてやアマチュアなのですから幅広い表現(本当になんの意味もない映像)も許容されるべきだと思います。一方向に閉じることのないように望みます。
鈴木 道明(25)埼玉県

かまた氏の意見は，あくまで1つの方向性を示しているにすぎません。賛同するも反発するもあなたしだい。

◆いやあ，8月号は「秋まつりPRO-68K」以来のよい特別企画だった。「秋まつりPRO-68K」のときは……ん？ あれ「SION IV」はまだ？ 2年もたったんだから，きっとすごいものができているんだろうなあ。
新谷 貴幸(20)埼玉県

ようやく，画面写真を撮れるところまでできてきた「SION IV」。とりあえずすごいかどうかは，SOFTWARE INFORMATIONの写真を見て，読者の皆さんが判断してくださいね。

◆X68000が熱暴走するのを見て，「ああ，オイルクーラーつけようかな」とか「本体にフィンを刻もうかな」などと考えてしまう私の頭も熱暴走中です。いやん。
八木沢 良二(21)栃木県

なんか，幸せそうでいいな。

◆編集部にはどんな感じの方々が，どんな場所でどんな様子で作業をしているのか，写真をつけてOh!Xで紹介してほしいです。
青田 秀樹(20)千葉県

ちなみにこのコメントを書いている人間は長髪野郎です。身の周りには，音楽CD，校正紙，週刊誌，パチンコ雑誌が散乱しています。というようなところなので，とても読者の皆さんにお見せできるような場所ではありませんから，かんべんしてね。

◆そろそろ新Xですね。すべてのパソコンがDOS/V化していくなかで，どのようなものが出てくるのか楽しみですね。
末吉 克行(27)兵庫県

◆とうとう新機種が出ましたね。デザインといい性能といい，シャープもやってくれたもんです。当然これから買いに行きます！ という文章を毎月送っていれば，この話題に関しては私が一番乗りになれるでしょう。でも，このハガキが掲載されるときには，発表されていることが望ましいですけどね。
片倉 純(20)宮城県

ここにいたって，なんだかじわじわと情報が出てきたという噂が……きちんとした情報であれば誌上で紹介しますので，楽しみにしていてください。

◆突如現れた「SX-WINDOW ver.3.1開発キット」の広告！ 発売日が書かれていませんが，すでに発売中なのでしょうか。それにしてもメディアがCD-ROMか……ドライブを買わなくてはならないな。ところで，これってGCCは入っているんでしょうかね？ ライブラリのみの配布？ 謎が多いですね。ところで，NetBSDもディスク本を出すようですね。これはCPUを68EC030から68030に変えないといけないのでしょうか。あと，「EX-System」の水玉模様が涼し気でナイスでした。
渡辺 浩史(24)秋田県

「SX-WINDOW ver.3.1開発キット」は，今月号に紹介記事が掲載されていますので参考にしてね。

◆そろそろX68000も本当に……と思っていたところにDSPボードに加えてSCSI-2ボードの発売。「EX-System」もそろそろ出そうだし，まだまだX68000も捨てたもんじゃないなと思いました。ところが9月号より再び値上げ……なかなか複雑なものを感じてしまいました。
宮坂 学(25)長野県

読者の皆さんがいるかぎりOh!Xは大丈夫……なはず。

◆あのう，アンケートハガキの「いちばんよかった記事」のところに「満開の電子ちゃん」と書いてもいいのでしょうか。この場合，よかったというよりも面白かったということですが。
高橋 秀之(33)東京都

……別にかまいませんよ。STUDIO Xの「恵子と学君」4コマが面白かったと書いてもね(すんすん，すねてやる～)。

◆7月中旬，大学で学部生の期末試験の試験官をやった。我が大学には，エアコンがないので見ているほうもそうだが，受けている学部生はもっと大変だ。でも受験者の行動を見ていると

なかなか面白い。さっさと問題を解いて退出する人。時間までがんばる人。あきらめて寝る人。解答用紙の裏にマンガを描いている人など……。なかでも驚いたのは，X68000の電卓を使っている人がいたことだった。吉池　信吾(24)東京都

そういえば，X68000を3台所有しているくせに，X68000の電卓は1個も持っていないな。せっかくだから電卓目当てにハガキを送ろうかな。

◆残暑が暑いざんしょ……スイマセン。さて，私は現在ナンバースに凝っています。お金も目当てですが，私は「2468」で勝負しております。それは，24→アルファベット24番目のXとあとの68でX68を意味するのです。別に68000でもよさそうなものですが，当たったならば周辺機器に突っ込みたいと思います(当分無理か)。

藤本　将景(25)高知県

ぜひ，当たるように祈っていますよ。

◆帰省する途中に日本橋に寄って，X68000のソフトを4本も買ってしまいました。新作が寂しくなったいま(電波までもが……)，とにかく買えるうちに買っておけをモットーに貧乏にあえぎながら，ゲーム三昧の日々を送っています。しかし，やり終えるのはいつの日になることか。

玉田　雄一(19)京都府

それでも終わらないことをどこかで祈っていませんか？

◆PDを買いました。とても満足しています。ハードディスクのバックアップも取れるし，Oh!X 8月号や電脳倶楽部のおかげで楽しい思いをしています。しかし，購入を考えていた「フリーソフトウェアセレクションvol.2」が売り切れてしまい，とても残念です。vol.3出ないかな。

多田　哲也(24)兵庫県

今度計測技研に打診してみましょう。

◆WINDOWSのMS明朝，MSゴチックをWF2VF.EXEで書体倶楽部に変更し，さらにVF2IFM.Xで半角フォントを作りました。これでようやく使えるフォントが手に入りました。

木越　英夫(36)愛知県

使えるものはなんでも使えの精神でこれからもがんばってくださいね。

◆Oh!Xを買ういつもの本屋にいつものように買いに行ってみると，先月まで1冊しかなかったOh!Xが突然3冊に増えていた。しかも，買ってからしばらくして，また売り場を覗いてみると，それが1冊に減っていた(つまりもう1人が買った)。売っているんだから売れるのは当然なんだけど，近くに別の読者がいるなんて嬉しい！

平間　大輔(18)北海道

ひょっとしてあと残り1冊が買われるまでそこにいたんじゃ……。

◆先日，大阪ソフマップのザウルス店で，「バーチャルボーイ」のお試しプレイコーナーがあったので遊んでみた。ちゃんと3Dで画面が表示されていてすごかったです。結構ヒットするかも?!

池田　國治(25)大阪府

最初は物珍しくて遊ばれていた編集部の「バーチャルボーイ」。最近は誰も触らなくなったなあ。確かに，面白い映像を見せてくれるのですが，異様に目が疲れるのはちょっといただけませんね。

◆以前から気になっていたのですが，ゲームの遊び方について，横内さんと西川さんの意見が分かれているみたいですね。一方は理想派，もう一方は現実派と，名指しを避けた2人の水面下の争いは，見事なまでに社会の縮図を誌面上に提供しているようですね。あと，これからのパソコンゲームのあり方についても十分興味をそそられますが，本筋から離れた中傷文より，互いに自刀を切りつけ合うような，相手の存在を認めた論争文を読んでみたいと思います。

鈴木　雄一郎(22)埼玉県

ゲームに対する鈴木さんの意見も聞きたいな。

◆新製品紹介のZipドライブですが，結構いけそうな気がしますね。安くて丈夫でわかりやすいというのは，大きなポイントでしょう。本当にメディアの容量とコストの改良，向上待ちってところでしょうか。さて，「ANOTHER CG WORLD」が終わってしまいましたが，本編の「響子 in CG わ〜るど」は終わらないでください。お願いします。　永井　邦彦(26)愛知県

メディアの大きさも結構重要なポイントですね。

◆最近は，Zipドライブ，PD，MO，MDなどの外部記憶装置がいろいろ現れましたね。今度，これらの長所，短所などをいろいろな方向からレポートなどしてほしいと思います。

小林　淳(26)埼玉県

気になる新製品としてJAZドライブなるものも出てきました。こちらも製品を入手次第レポートしていく予定ですので，楽しみにしていてください。

◆SCSI-2はほしいけど，クロックアップに対応していないのはきつい。CZ-6VS1のためにも絶対必要なんだけど，アニメーションの再生には10MHzじゃきついし，かといってX68030を買うお金もない(DSPボードは買ってしまった)。タイミングの問題かどうか詳しいことはわからないけど，ほとんど一般化してしまったクロックアップには対応してほしかったな。とりあえず，自分でできること(DSPとかVSIのプログラミング)をしているしかなさそう。ハードウェアはよくわからないから他力本願モードなんだよね。誰か〜，クロックアップに対応したSCSI-2ボード(転送が速ければなんでもいい)を作って〜。

若月　功(21)埼玉県

世の中，そううまく動かないので，自分自身でがんばりましょう。

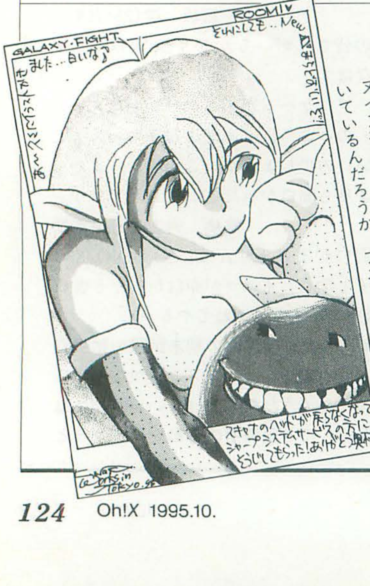
◆パソコンのためにと用意した金が自転車になってしまいました。目標はツール・ド・オキナワ。「なんと無謀なことをしているんだろう」と思っている自分です。が，「たまにはいいかもしれない」と開き直っています。11月に出場しますが，幸運祈ってください(珍しく自転車に金をかけたら，成りゆきでこうなっちゃいました)。

藤原　彰人(25)岡山県

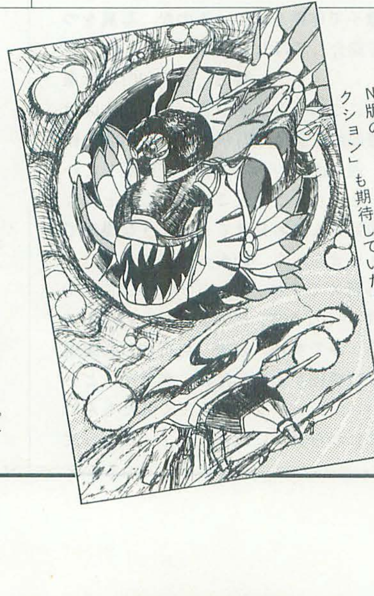
人生いきあたりばったりです。自分に正直に生きるのが一番。ということでがんばってね。

◆やはり夏になると暑いですね。学校にX68000を持ち込んどいて正解でした。なぜなら冷房が効いているから(夜中も)。学校で生活して，家へは寝るのと風呂へ入るために帰るだけのような生活をしています。　西尾　昌人(21)愛知県

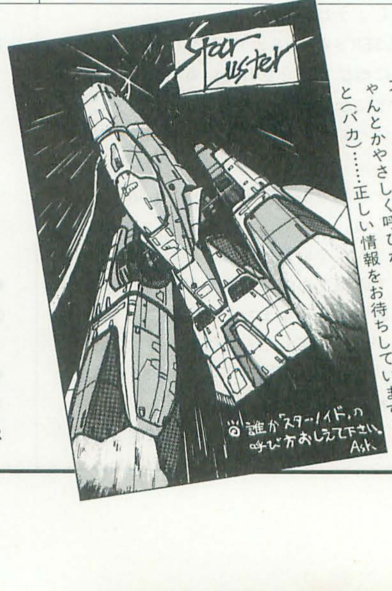
この夏の猛暑はハンパじゃなくすごかったですね。夜行性の人間としては，本気でつらいかったなあ。



▲中川　和之　埼玉県
う〜んイヌイヌ手袋(勝手に命名)がかあいい。イヌイヌ手袋ということは，ニクキュウ(肉球)がついているんだろうか。プニプニしたいなあ。



▲松居　啓樹　富山県
シューティング好きの私は，SEGA SATURN版の「ダライアス外伝」とともに「レイヤーセクション」も期待していたりします。



▲上田　考一　広島県
スターノイドの呼び方ですか。やっぱり，ノイちゃんとかやさしく呼びかけるといいんじゃないかと(バカ)……正しい情報をお待ちしています。



▲清家　亜紀　福岡県
教育実習の相手は小学一年生とのこと。やんちゃ盛りのお子様方に清家亜紀が立ち向かう！　迫真の実習レポートを期待してます。



▲今井　健生　奈良県
「そして晴れて自由の身」ということですが，それも入社までのわずかな間です。つかの間の自由人を堪能してくださいね(意地悪〜)。



▲岩瀬　貴代美　福岡県
風の精に願いを込めて……といった感じのさわやかなイラスト。日々生活に追われる現実から離れてこんな空想世界に思いをはせるのもいいかもね。

◆SEGA SATURNをRGB化したのですが，走査線の隙間が気になるので，インタレース化をしました。ただ，きれいになったのはよかったのですが，気がつかない間にただでさえ悪い視力がさらに低下してしまいました。

中島　貴史(19)滋賀県

ゲームは1日1時間まで。この原則を守らなきゃ。

◆7月13日。初めてヘリコプターに乗りました。名目上，山奥に残された人たちの救出ということだったのですが，食料の余裕とX68000を動かす電気が止まらなかったので，あまり救出された気がしません。でも，この大雨でトロッコ電車の線路が埋まったり，下界との連絡がつかなかったりで年内の復旧は無理だとか，ボーナスが出ないとかの噂が……。

黒田　博明(25)富山県

あまり深刻な状況にならないといいですね。

◆就職先を探さなくてはならないのに「PICTパズル」が熱い。　木下　与之(19)石川県

遊び飽きたらオリジナル問題の作成に挑戦してみよう。面白い問題ができたら投稿よろしくね。

◆とある商店の自動販売機に「なにが出るのかお楽しみ」コーナーというのを見つけた。私は「お楽しみ」というより「不安」になり，その晩はまんじりともしなかった。

大久保　明弘(23)岩手県

結局余ったジュースを詰め込んでいるだけでしょう。ホットなコーラや缶入り味噌汁など入っているわきゃ……さあ，大久保さん，不安を押し退けて挑戦するのだ！

◆先日，いきつけの某ゲームコーナーにて浪人生活を享楽(おい！)しているときに，ひたすら車ゲーで遊んでいる人を発見。しばらく観察していると，その人はタイトーの「SUPER CHASE」(古いけど私も好き)で，15コインクリアを達成。そのとき私は，「人間やればできる」ってことを痛感した(笑)。でもよく考えてみると，その人は，あらかじめ15枚以上100円玉を用意していたんですね。このことに気づいたとき，私はさらなる感動のあまり，2本指を天に突き刺してグルグルしちゃいました(ウソです)。

平野　鉄之助(19)長野県

清く正しい浪人生活を謳歌しているようでなによりです。

◆私のバイト先では，いつのものかはわからないが，30Mバイトのハードディスクがいまだに動いている。ゆえにいいたい。「いまだにMC68000の10MHzのマシンを使ったっていいじゃないか」

真田　知之(23)北海道

関係ないけどコダックの「見かけで選んでなにが悪いの！」というCMは，個人的にちょっと嫌いです(本当に関係ない話でやんの)。

◆やり残していることは山ほどある。このまま死んじまうわけにはいかんのだ。

東　裕人(24)島根県

やりたいことが山ほどあるということはいいことです。がんばってくださいね。

ぼくらの掲示板

●掲載ご希望の方は，官製ハガキに項目(売る・買う・氏名・年齢・連絡方法……)を明記してお申し込みください。
●ソフト・コンピュータ本体の売買，交換については，いっさい掲載できません。
●取り引きについては当編集部では責任を負いかねます。
●応募者多数の場合，掲載できない場合もあります。
●紹介を希望されるサークルは必ず会誌の見本を送ってください。

仲間

★X680x0のディスクマガジンサークル「MeccaTick」(メカティック)では，活動活性化のために読者・参加者を募集しています。コンセプトは「配布を受けてくださる方が気軽に参加できる場所」というもので，内容に関しては各自個人に任せています。興味のある方は190円切手を貼った返信用封筒と5インチフロッピーディスク4枚を同封して下記の住所までお送りください。〒221　神奈川県横浜市神奈川区神大寺1-4-5　双葉荘105　小田原　真之(19)

★X6800x0ユーザーのためのサークル「MANATEE」では，活動活性化のために会員を募集します。当サークルでは，音楽，フリーソフトなどを収録したディスクマガジンを不定期で発行しています。フロッピーディスクを通してのコミュニケーションに興味をおもちの方は，切手を貼った返信用封筒と2枚の5インチフロッピーディスク(3.5インチは不可)を同封のうえ，下記の住所までお送りください。〒319-15　茨城県北茨城市磯原1965　野口　友則

売ります

★X68000XVI用2MバイトRAMボード「CZ-6BE2A」を13,000円(送料別)で売ります。箱，説明書あり，完動品です。連絡は往復ハガキでお願いします。〒630-02　奈良県生駒市谷田町1275-40　浅利　宗弘

編集室から from E・D・I・T・O・R

DRIVE ON

このコーナーでは，本誌年間モニタの方々のご意見を紹介しています。今月は8月号の内容に関するレポートです。

●今回の付録ディスクの感想ですが，一番の目玉はCPKPLAY.Xです。珍しくDOS/Vユーザーの知人を唸らせることができました。それにこれ目当てにCD-ROMドライブを買ってしまった人もいます。恐るべし「ＰＶ」。

X68000というとすでに時代遅れというイメージが自他ともに定着しつつある現在，最新ゲーム機と同等のことができてしまうCPKPLAY.Xはとても魅力的なプログラムです。結局は他機種のデータの流用なのですが，各種データの再生マシンというのも立派なX680x0の生き残る道だと思います。以前，友人との会話で「グラフィック閲覧用だけにでもX68030を買っておきたい」という結論になり，これからはそうした人が増えるかもしれませんね。

もうひとつ，EX-Systemについてですが，そのルーツはZ's staffでPIC形式のセーブ，ロードを可能にするPicfilerにあったと思います。それがいまでは独立して動くグラフィックシステムになったのですからなんともすごい話です。とりあえず触ってみた感想ですが，パターンブラシが面白かったです。特にシャボン玉のパターンは「まるで少女漫画の世界だ」と思いながら，画面をパステルカラーのシャボン玉で埋めつくしてしまいました。放射光も結果はきれいで面白いのですが，いかんせん時間がかかりすぎます。10MHz機では冗談ではないでしょう。製品版ではX68030＋68882やXellent30に対応したものも収録されるそうですが，10MHz機＋68881にも当然対応してくれるんですよね。

中村　健(25)　X68030, X68000 ACE-HD, PC-386GS, MSX2＋　埼玉県

●バランスのよい付録ディスクといえば聞こえはいいが，使う気になったのはZ-MUSICだけ。確かにグラフィック，音楽，ゲームなど，いろんな人を対象にしていたが，私は結局興味のあるものしか使わなかった。それぞれが比較的大きな容量だけに，バランスがよい分，使えるものが少なかった気がする。もう少し小物のツールを増やしてもよいのでは？

北浦　暁光(21)　X68000 XVI-HD, X1 turboZ III, PC-8801mkIIFR　東京都

●８月号では２枚のボードが紹介されていました。まず，SCSI-２ボードですが，X68000でそれほど騒がれなかったのは，よくいえばCPUパワーとSCSIボードのバランスが取れていた，悪くいえばその必要性がないほどマシンが遅いといったところでしょうか。実際はFDユーザーが多いためディスクのアクセスを最小になるように作成されたソフトが多く，SCSIの速度が全体の速度に与える影響が少なかったということでしょう。

「Mach-２」のレポートを読むと，私がPC-98で体験したときの興奮が思い出されました。そうです，CPUパワーよりもディスク速度に敏感になるべきなのです。より多くの人がそれに気づいてくれることを願っています。

X68030でBSDを使う人も増えてきました。UNIXのようなOSではディスクの速度が全体のパフォーマンスに大きく影響します。それだけにBSDで使用した感想なども加えていただけると助かる人もいたのではないでしょうか。

あと「AWESOME-X」についてですが，これはマニアックですね。X68000では，もともと少ないボード類に，「V70ボード」や「Polyphon」など変わったボードがけっこう存在していました。それに，X68000で発売されるボードはユーザーからの強い要求があって発売に至ったというものは少なく，ボードを企画する側が自己満足のために作ったと思えるようなものが多い気がします。

話がそれましたが，DSPボードには非常に興味があります。以前，Macintoshで「AudioMedia」というシステムを使ったことがあり，DSP56000というDSPが使われていました。「AWESOME-X」で使われているTIのDSPはそれよりもメジャーで，資料も豊富にあり，使いこなせる人も結構見かけます。

レビューの中でアセンブラが付属していることを伝えている点は，当然ですが評価できます。TIのDSPに触れたいけれども，そのための環境がないという人に，X68000はお役に立てるのです。

しかし，「あれをやろう」という目的達成だけを目指す人に対して敷居が高いことも事実です。目的達成に加えて，達成への過程を楽しめる人に向けて用意されているというのは，X68000らしくていいと思うのですが，いかがでしょう。「Your Own Risk」という言葉，私は大好きです。

浅野　憲(24)　X68000 PRO, Macintosh Centris650, PC-98RL, X1F, MZ-80Ｃ, AppleII, PC-1600Ｋ, PC-1245　東京都

●満開製作所から発売されたSCSI-２ボード「Mach-２」ですが，８月号のレビューを読むまではX68000のSCSIインタフェイスが遅いなんて考えもしませんでした。これは単に私のマシンが遅いからでしょうか。なにはともあれ，速いことは悪いことではないので，現状のスピードにイライラしていた人にとっては朗報でしょう。あと，「AWESOME-X」は私には高速シリアルボードのような感覚でしかありません。今後のソフトのサポートに期待ですね。私はIRポートでザウルスと……などと考えてしまいますが。

大上　幸宏(22)　X68000 PROII　鹿児島県

●PDよりもZipのほうに期待が大きいですね。フォーマットして95Ｍバイトということですが，十分実用になると思います。やはり，速さは力！　狭いといわれても，高速であることが大切だと思います。リムーバブルメディアで大きさも3.5インチMOくらいと，もち運びも苦になりませんし。容量については，「大きくなる」と誰もが思っているらしいので，大きくなるのでしょう。

あと，耐久力の問題があります。しかし，なんともおかしな方法で読み書きをするようなので，かなり長期間使用できるとは思いますが，その点をレポートしてほしかったです。それに，パワースイッチがついていないのは……。つけてほしいなぁ。

三隅　信幸(22)　X1turboZII　広島県

ごめんなさいのコーナー

7月号　ショートプロぱーてぃ

P.77　BDSELECT.RでEXCONFIGの設定に誤りがありました。正しくは，

EXCONFIG＝\SYS\BDSELECT.R \SX \GRAPHIC \MUSIC \WP \ETC

です。ご迷惑をおかけしたことをお詫びいたします。

9月号　Oh!X LIVE in '95

P.66　「ときめきメモリアル」より「告白」のライセンス標記が間違っていました。正しくは© 1994 KONAMI All right reservedです。関係者の方々に大変ご迷惑をおかけしました。お詫びいたします。

バグに関するお問い合わせは
☎03(5642)8182(直通)
月～金曜日16：00～18：00

お問い合わせは原則として，本誌のバグ情報のみに限らせていただきます。入力法，操作法などはマニュアルをよくお読みください。

また，よくアドベンチャーゲームの解答を求めるお電話をいただきますが，本誌ではいっさいお答えできません。ご了承ください。

画面で見るだけじゃもの足らない いざ，出力せん

▶1991年8月号以来の印刷関係の特集です。さすがに4年も経つと，パソコンの性能がどんどん向上していくように，プリンタの性能も向上してきました。いまでは，かなり安く，高品質な出力ができるプリンタが登場してきました。しかし，本体の進歩以上に重要なものがあります。それはドライバ(印刷プログラム)です。X68000は市場という観点から見ると取り残されたといえる状況です。そんな中，メーカーからのサポートは期待できません。結局，プリンタの性能を引き出すためにはドライバを自前で作成するしかないのです。逆にそんなプログラムを作れるユーザーがいるということが素晴らしいのですが。

プリンタといえば，いまはカラーインクジェットが主流です。しかし，熱転写型のものもかなり高品質なものが登場してきました。これも機会があれば紹介してみたいですね。

もちろん，印刷といっても，多くの人の場合は，文字がメインになるでしょう。「シャーペン」を使った文字出力も特集で取り上げていますので参考にしてください。ただ，残念なことに，今月号が出る頃には「シャーペンワープロパック」の新しいバージョンが登場しそうなのです。これには新しいプリンタドライバも入っているようなので，期待をしたいところですね。手に入り次第レビューをお届けする予定です。楽しみにしていてください。

▶今月号で紹介したNECのモニタですが，画面撮影の半数くらいに使用しました。わかりますか？　すでに，X68030が発売されてからも，すでに2年半が経ちます。メガディスプレイ計画を実行した人もそうでない人も，ディスプレイがかなりくたびれてきているのではないでしょうか？　モニタを買い換えるときには，考慮に入れても問題ないモニタでしょう。色が黒でないのが残念ですが。

▶今月は，予告どおり「DōGA CGアニメーション講座」，著者多忙のため「ハードコア3Dエクスタシー」などがお休みです。

投稿応募要領

●原稿には，住所・氏名・年齢・職業・連絡先電話番号・機種・使用言語・必要な周辺機器・マイコン歴を明記してください。

●プログラムを投稿される方は，詳しい内容の説明，利用法，できればフローチャート，変数表，メモリマップ（マシン語の場合）に，参考文献を明記し，プログラムをセーブしたフロッピーディスクを添えてお送りください。また，掲載にあたっては，編集上の都合により加筆修正させていただくことがありますのでご了承ください。

●ハードの製作などを投稿される方は，詳しい内容の説明のほかに回路図，部品表，できれば実体配線図も添えてください。編集室で検討のうえ，製作したハードが必要な場合はご連絡いたします。

●投稿者のモラルとして，他誌との二重投稿，他機種用プログラムを単に移植したものは固くお断りいたします。

あて先

〒103 東京都中央区日本橋浜町3-42-3
ソフトバンク出版部
Oh!X「テーマ名」係

SHIFT・BREAK

▶divを作るのにしばらくC＋＋ばっかり使ってたら，Cで書くのさえつらくなってしまった。ほぼすべてアセンブラで書いている某フリーソフトに手を入れるときなんて，アセンブラで書く気は毛頭なくてCで書いてみたらリンクがどうしてもできない。結局コンパイラにアセンブラソースを吐かせて，リンクできるように手で修正した。　(けんと)

▶1年ぶりに上京してみると，家への帰り方を思い出せなかった。4年目にしてやっと札幌に慣れて堕落したのだろうか。とにかく，東京が非人間的な環境であることに気づいてしまったのだ。東京の思い出は悪しきものとして抹消したのだろう。こんなアナーキーな都市，日本の恥部だ。と思いつつも，やっぱり東京に出ないとならないのは腹立つぜ。(威)

▶コミケでは多忙につき，オリジナルテープを落としてしまいました。何人か，見にきてくれましたが，「テープ落とした」という貼り紙を見て，帰っていく様子。心苦しかったです。次回は必ずなんとかします。ごめんなさい。しかし，今月，生きてるのが不思議かも。ライターやってて，初めて「失踪」したいと思っちゃった。　(瀧)

▶去年の11月に買ったカラーノートの液晶が腐り出した。妙に色あせて，バックライトを最低にしないとよく見えない。電源を入れても数分間つかないときもある。メーカーの保証書がなくて安かったのだが，販売店の1年保証はついているので，もうちょっと経って，完全に型落ちになってからもっていけば，新型と交換してくれるかも!?　(I.K)

▶数年前に始めたリバーツーリング。折り畳み式のカヤックやシュラフなど40kgほどの荷物を背負って，キャンプをしながら川を下るのだ。去年のいまごろは，水の少ない長良川で悪戦苦闘していたけれど，今年は秋口の晴れ間をねらって2，3本の川を下ろうかと思っている。それとは関係ないが，タレントってどうして日常もタメロなんだろう。(ats)

▶実は最近，ちとwww巡りに凝ってたりする。あ，サーフィンっていうんだっけ？　しかし，夜はどこのプロバイダも犬かき以下の遅さ。昼の3倍くらい時間がかかる。これでテレホーダイ(夜11時以降朝まで電話かけ放題料金制)なんか使った日には，睡眠不足で倒れちゃうわさ。むむむ。　(で，朝は新聞代わりに仕事場でhttp://mainichi.co.jp/と☆(で))

▶Londonの本屋で日本の漫画を英訳して掲載している「MANGA」という雑誌を発見。「バキッ!!」のような「効果音系文字」の処理が困難でトレーシングペーパーで再描画したのだろうか？　線のふらついた大雑把な絵がページ一面に広がっていた。背景の描き込みの激しい知っている漫画も載っていたのだが，ほとんどのコマの背景が白だった。合掌。　(善)

▶ある日突然ラインプリンタ（BJ）に我慢できなくなり，レーザープリンタを買いに走った。最近は安くなったもので，ハードだけなら実売6万円を切っている。とはいえPostScript対応であることが仕事上必須。対応品は値段が倍になるが，それでも十分安い。レーザープリンタの印刷はエッジがくっきりとしてにじまない。幸せだ。　(A.T.)

▶PATに当たった。電話回線を通じて馬券が買えるというものだ。その購入に使える機器はファミコン(スーファミではない)，専用電話(ディスプレイフォンでファミリーマートに置いてあったような……)，パソコンなどがある。早速，同居人が所有しているPC-9801DAを奪取。動作環境はDOSかWindows3.1。で，通信速度は……1200bps……ガーン。　(高)

▶「SION IV」開発のため，自宅にX68000 Compact XVI＋MO＋6Mバイトメモリ一式を常駐させた(また金が飛んだ)。会社でポチポチ，家でもポチポチとキーボードを打つはめに……まあ，最後の追い込みだからしようがないけど，気分はもう世捨て人。ときたまアミュレット探しの旅に出ちゃうけど，きっと12月号には間に合うのでしょう。　(J)

▶長年探していたCDが昨年末からこの夏にかけて軒並み発売されていたことがわかった。オザワのチャイコフスキー5番の3録音とミュンシュのベートーベン9番なのだが，やはり細かくチェックしとかなければいかんなあ。しかし，ロマン派以降は作曲者と指揮者が同じ国の人でないほうがよいと思うのは私だけだろうか。　(U)

▶8月24日に，Windows95(英語版)が発売された。日本語版も3カ月後に発売の予定で，今後1年間の販売見通しは全世界で1億コピーとか。すごいねえ。米国じゃ新聞はどこも1面を飾る騒ぎよう。もはや国民的な関心事なのかと思いきや，テレビの街頭インタビューでは，Windows95？　なにそれ？　って感じの人も。ちょっとホッとしたな。　(T)

microOdyssey

Xシリーズといえば NEC製品のことを意味することのほうが多いかもしれない今日この頃。ようやく新機種（シャープ製）の話題もちらほら挙がるようになってきた。噂レベルの情報は多いのだが，完成形はまだ見えてこない。とりあえず開発しているというのは確実だ。

多くの人は初代X68000が登場したときのような劇的なスペックアップを望んでいると思われるが，だいたい感触としてはある程度それに匹敵するものになると思われる。しかしいまの世の中，独自のハードウェアというものが可能だろうか？

現在ではMacintoshもAT互換機もCPUが違うだけで構成はほとんど同じだ。ちょっとアレンジすればPC-9821にもなる。CPU，メモリ，あとはPCIバスの下に周辺をまとめてできあがりだ。この構成ではなにをぶら下げるかというのはそれほど大きな問題ではない。

現状のマシン分類は，若干の機種を除けば，Macintosh，AT互換機，ワークステーションといった分け方にしかならない。CHRPマシンが現れたとしても，結局はMacintoshか，ワークステーション，またはWINDOWSマシンの分類になってしまうだろう。これはすなわち，MacOSが走るか，WINDOWSが走るか，UNIXが走るかといったOSによる分類にほかならない。

ハードウェアによる個性というものは徐々になくなりつつある。

こういったPCIマシンのハードの作り方自体はそうおかしなものではない。周辺も極度に高速化されていく。CPUは一昔前には考えられなかったくらい高速である。どの機種でもなんでもできるようになったというのは，ある意味で事実であろう。にもかかわらず，それらのマシンが実現しようとしている「マルチメディア」環境はそういった重装備を嘲笑うかのように，さらに高速なデバイスを要求している。

CD-ROMから動画は再生できる，ただし小さな画面がカクカクと動く。本体だけでMIDIデータも演奏可能だ。しかし，演奏中は処理速度が極端に遅くなる。コンピュータ本体はもの凄く高性能にはなっても，実現しているもの自体はそれほど高度なものには思えないのだ。数年前に提案された環境がこれだけの資源を投下してようやく動きだしたといったところだろう。

1，2年後の世界ではCPU自体はいまの2倍速くなる。最新ゲーム機程度のポリゴン処理は苦もなく処理するボードも低価格で発売されるだろう（すでにそのようなチップは発表されている)。「マルチメディア」といった言葉がまたまた新しい意味を加えていくだろう。そんな状況でどのような環境が実現されていくのか……というと，あまり見えてこない。現状ではベースになるOSやユーザーインタフェイスがどうしようもなく古くさいのだ。これらはハードほど進化が速くない。

画一的なハードと旧態依然のOS，これでは個性的なマシンは生まれない。

思えば，ちょうど私がこの会社に入ってきたのはX68000前夜ともいうべき時期だった。当時の編集長からMPU68000とワークステーションの勉強をしなさいといわれた日のことは忘れられない思い出である。そして，そろそろ入社10年になろうとしているのだが……。　(U)

1995年11月号10月18日(水)発売

特集　はりきって使うCD-ROM

・データ解析の勘所
・とにかくデカいBMPを読む

新製品紹介
シャーペンワープロパック ver.2
全機種共通システム
S-OS版PICTパズル

バックナンバー常備店

東京	神保町	三省堂神田本店5F	03(3233)3312
	〃	書泉ブックマートB1	03(3294)0011
	〃	書泉グランデ5F	03(3295)0011
	秋葉原	T-ZONE 7Fブックゾーン	03(3257)2660
	八重洲	八重洲ブックセンター3F	03(3281)1811
	新宿	紀伊国屋書店本店	03(3354)0131
	高田馬場	未来堂書店	03(3209)0656
	渋谷	大盛堂書店	03(3463)0511
	池袋	旭屋書店池袋店	03(3986)0311
	八王子	くまざわ書店八王子本店	0426(25)1201
神奈川	厚木	有隣堂厚木店	0462(23)4111
	平塚	文教堂四の宮店	0463(54)2880
千葉	柏	新星堂カルチェ5	0471(64)8551
	船橋	リブロ船橋店	0474(25)0111
	〃	芳林堂書店津田沼店	0474(78)3737
	千葉	多田屋千葉セントラルプラザ店	043(224)1333
埼玉	川越	黒田書店	0492(25)3138
	川口	岩渕書店	0482(52)2190
茨城	水戸	川又書店駅前店	0292(31)0102
大阪	北区	旭屋書店本店	06(313)1191
	都島区	駸々堂京橋店	06(353)2413
京都	中京区	オーム社書店	075(221)0280
愛知	名古屋	三省堂名古屋店	052(562)0077
	〃	パソコンΣ上前津店	052(251)8334
	刈谷	三洋堂書店刈谷店	0566(24)1134
長野	飯田	平安堂飯田店	0265(24)4545
北海道	室蘭	室蘭工業大学生協	0143(44)6060

定期購読のお知らせ

Oh!Xの定期購読をご希望の方は綴じ込みの振替用紙の「申込書」欄にある『新規』『継続』のいずれかに○をつけ，必要事項を明記のうえ，郵便局で購読料をお振り込みください。その際渡される半券は領収書になっていますので，大切に保管してください。なお，すでに定期購読をご利用の方には期限終了の少し前にご通知いたします。継続希望の方は，上記と同じ要領でお申し込みください。

基本的に，定期購読に関することは販売局で一括して行っています。住所変更など問題が生じた場合は，Oh!X編集部ではなくソフトバンク販売局へお問い合わせください。

海外送付ご希望の方へ

本誌の海外発送代理店，日本IPS(株)にお申し込みください。なお，購読料金は郵送方法，地域によって異なりますので，下記宛必ずお問い合わせください。

日本IPS株式会社
〒101 東京都千代田区飯田橋3-11-6
☎03(3238)0700

Oh!X 10月号
■1995年10月1日発行　定価760円(本体738円)
■発行人　橋本五郎
■編集人　稲葉俊夫
■発売元　ソフトバンク株式会社
■出版事業部　〒103 東京都中央区日本橋浜町3-42-3
Oh!X編集部　☎03(5642)8122
販売局　☎03(5642)8100 FAX 03(5641)3424
広告局　☎03(5642)8111
■印刷　凸版印刷株式会社

満開の電子ちゃん

作 中村隆生（なかむら たかお）
え 岡村祭（おかむら まつり）

パソコンショップ満開の電話は03(3985)5530です。



ホントに事務所を移転しました。〒171東京都豊島区西池袋5-17-11ルート西池袋ビル901号。
電話03(3985)6110 FAX03(3985)5366です。

講読方法：定期購読、ソフトベンダーTAKERU、NIFTY-SERVEでお買い求めいただけます。
また、JCB、VISAカードもご利用になれます（金額9,000円以上の場合）。
★定期購読（送料サービス、消費税込）3ヶ月＝4,500円、6ヶ月＝9,000円、12ヶ月＝18,000円。
・現金書留：〒171 東京都豊島区西池袋5-17-11 ルート西池袋ビル901 ㈱満開製作所
・郵便振替：02810-13298 口座名 電脳倶楽部
・JCB・VISAカード：フリーダイヤル0120-887780または、NIFTY-SERVE GO MANKAI。
ご注文の際には、郵便番号、住所、氏名、電話番号、タイプ（5インチ・3.5インチ）、新規購読か継続購読かを必ずお知らせ下さい。新規購読の際、購読開始号のご指定のない場合は既刊の最新号よりお送りいたします。製品の性格上返品には応じられませんが、お申し出があれば定期購読を解約し残金をお返しいたします。
★TAKERUでお求めの場合、75号までは1,200円（税込）、76号以降1部1,600円（税込）です。
★お問合わせ先 TEL03-3985-6110（月～金 午前11時～午後6時）。
★バックナンバーは創刊号よりございます。★フリーダイヤルは、月～金 午前10時～午後5時。

私はまだ十九歳。でも主婦。主婦ったら主婦なの。電脳倶楽部を読みはじめてまだ一年。主人が読んでいるのを見て一緒に見るようになった。最初は主人のことオ・タ・クって思ったけど、私もハマってからは主人が毎月楽しみにしている理由がわかった。「電源オンですぐ起動、マウスひとつでらくらく操作」…この言葉どおりパソコンなんて全然わからない私でも今は電倶オタク。そう私は主婦。主婦ったら主婦なの。うーん、私はビープ音と画廊が大好き！ あなたも早く定期購読をはじめてはい・か・が。



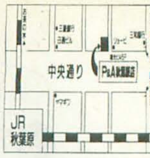
（京都府）谷口小百合



(秋葉原店は来店のみ、通信販売は行っておりません.!!)

P&A 秋葉原店オープン

営業時間●AM11:00〜PM7:00(日祭PM6:30)(第三水曜定休)
TEL.03-5294-7053 FAX.03-5294-7054
通販部=本店(新小岩) TEL.03-3651-0148 FAX.03-3651-0141



注目!!冬のボーナス一括払い手数料(金利)無料(平成7年10月末/11月末/12月末のいずれかをご指定下さい。)

X68030お買い得セット

(クレジット表:送料、消費税込み)

①ハードディスクセット

- ●CZ-500C(本体)
- ●340MB(外付)ハードディスク

定価¥506,000
P&A超特価 ¥218,000
(ハードディスク540MBに変更の場合¥5,000加算して下さい。)

12回	19,800	24回	10,400	36回	7,200
48回	5,600	60回	4,700		

②モニターセット

- ●CZ-500C(本体)
- ●CZ-608D-B(モニター)

定価¥492,800
P&A超特価 ¥255,000

12回	23,100	24回	12,100	36回	8,400
48回	6,600	60回	5,500		

■②のモニター変更の場合
- ●CZ-615D(チューナー付)に変更の場合 ¥56,000
- ●CZ-621D(B)……………に変更の場合 ¥64,000

加算して下さい。

X68030/68000オリジナルセット

⊙CZ-500C	⊙CZ-500C	⊙CZ-674C	⊙CZ-674C
●HD(内蔵)500MB ●メモリー8MB増設(合計12MB) ●コプロ ●SX-WIN インストール済み	●HD(内蔵)800MB ●メモリー8M増設(合計12MB) ●コプロ ●SX-WIN インストール済み	●HD(内蔵)500MB ●メモリー6M増設(合計8MB) ●SX-WIN インストール済み	●HD(内蔵)800MB ●メモリー6MB増設(合計8MB) ●SX-WIN インストール済み
特価 ¥288,000	特価 ¥315,000	特価 ¥176,000	特価 ¥199,000

⊙内蔵ハードディスク(500C、674C用)(単品)
●500MB…特価¥49,800 ●800MB…特価¥69,800
(当社取り付けの場合、¥8,000加算して下さい。)

決算大処分セール 旧シリーズ今が買いどき!!

(送料¥2,000・消費税別)(クレジット表:送料・消費税込み)

X68000 Compact XVI

- ●CZ-674C-H
- ●CZ-608D(B)

定価¥392,800
P&A特価 ¥134,000

12回	12,300	24回	6,400	36回	4,500	48回	3,500	60回	2,900

- ●CZ-674C-H
- ●CZ-608D(B)
- ●CZ-6FD5

定価¥492,600
P&A特価 ¥182,000

12回	16,600	24回	8,700	36回	6,000	48回	4,700	60回	3,900

決算大処分セール 旧シリーズ今が買いどき!!

(送料¥1,000・消費税別) 単品、限定

⊙PROII-HD
●CZ-663C
(ハードディスク40MB内蔵)
P&A超特価 ¥39,800

⊙PROII-HD 最強モデルセット
- ●CZ-663C
- ●メモリー11MB増設(合計12M)
- ●SCSIボード付

P&A超特価 ¥119,000

⊙Compact XVI
●CZ-674C
P&A超特価 ¥76,500

X68用専用ディスプレイ

- ⊙CZ-608D-B 特価¥59,800
- ⊙CZ-615D 特価¥118,000
- ⊙CZ-621D 特価¥120,000

MIDIセット

- ●MC-6600(SNE)
- ●SX-68MII(システムサコム)
- ●MIDIケーブル

特価¥45,800

- ●SC-55MKII(ローランド)
- ●SX-68MII(システムサコム)
- ●MIDIケーブル

特価¥56,800

(SC-88に変更の場合¥29,000加算して下さい。)

単品
- ●MC-6600(SNE)…………特価¥32,800
- ●SC-55MKII(ローランド)·特価¥43,600
- ●SC-88(ローランド)………特価¥73,500
- ●SC-88VL(ローランド)…· 特価¥54,500

スピーカー

- ●SP-300(シグマ)……特価¥ 4,980
- ●SC-C55(AIWA)………特価¥ 5,980



ALTEC ACS300 特価¥37,000
ALTEC ACS100 特価¥16,000
YAMAHA YST-M5 特価¥6,400

X68000/68030用 メモリボード (送料¥700・消費税別)

■I/Oデータ
- ●SH-5BE4-8M(30用)…·特価¥39,500
- ●SH-6BE1-1ME(600C用)…特価¥10,200
- ●PIO-6BE1-AE(ACE/PRO/PROII用)·特価¥10,200
- ●PIO-6BE2-2ME(拡張スロット用)·特価¥19,600
- ●PIO-6BE4-4ME(〃)·特価¥33,600

■シャープ
- ●CZ-5BE4(30用)………特価¥39,800
- ●CZ-5ME4(5BE4用増設)·特価¥36,500
- ●CZ-6BE2A(XVI用)……特価¥38,900
- ●CZ-6BE2B(XVI、674C増設)特価¥37,500
- ●CZ-6BE2D(674C用)…特価¥20,500

モデム&FAXモデム (送料¥1,000)

<アイワ>
- ●PV-BF144(ボックス型)……特価¥17,000
- ●PV-AF288(推奨機種・XVI以上)……特価¥32,000

<マイクロコア>
- ●MC144FXe/w(ボックス型)……特価¥14,800

<オムロン>
- ●ME1414BII(ボックス型)……特価¥17,000
- ●ME2814B(推奨機種・XVI以上)……特価¥29,800
- ●MD-144XT10V(限定在庫限り)……特価¥30,000

MO (送料¥1,000)

Logitec
- ●LMO-200(128M)……定価¥69,800▶特価¥45,800
- ● 340(128M)……定価¥79,800▶特価¥52,300
- ● 400(128/230M)定価¥118,000▶特価¥76,000
- ● 420(230M)…·定価¥138,000▶特価¥99,000

ICM
- ●MO-120S-N…………定価¥74,800▶特価¥55,000
- ● 230S-N…………定価¥118,000▶特価¥87,000

Filo
- ●CS-M230PA(230MB)定価¥148,000▶¥77,800

CD-ROM (送料¥1,000)

Logitec
- ●SCD-200(2倍速)……定価¥19,800▶特価¥16,800
- ● 420(4倍速)……定価¥34,800▶特価¥27,200
- ●LCD-440(4倍速)……定価¥39,800▶特価¥26,800

ICM
- ●CD-620S-N(4倍速)·定価¥34,800▶特価¥26,400

緑電子
- ●CXA-660-98(4.4倍速)定価¥39,800▶特価¥33,200
- ● 660-5L(〃)定価¥49,800▶特価¥44,200

東京システムリサーチ製(X SIMM)

(送料¥700・消費税別)

(X SIMMVI)
⊙XVIシリーズ専用SIMM増設式メモリボード
- ●X SIMMVI(634C用)…定価¥16,500➡特価¥13,000
- ●X SIMMVIc(674C用)·定価¥16,500➡特価¥13,000

⊙増設SIMMメモリ(72PIN)
- ●4MB(70ns)……………特価¥11,800
- ●8MB(70ns)……………特価¥27,800
- ●4MB(60ns、24MHz以上用)………特価¥16,500
- ●8MB(60ns、24MHz以上用)………特価¥28,000

- ●6MB(60ns、メーカー純正品)……特価¥27,800

(X SIMM 10) ⊙SIMM増設式メモリボード
- ●X SIMM 10………定価¥18,000➡特価¥15,700

⊙増設SIMMメモリ
- ●1MB×2……特価¥10,000
- ●4MB×2……特価¥30,000
- ●10MB例 X SIMM10+1MB×2+4MB×2··¥55,700

X68000/68030専用ハードディスク (送料¥1,000・消費税別)

外付

■ジェフ
- ⊙GF-340(330MB、13ms)…………特価¥24,800
- ⊙GF-540(520MB、12ms)…………特価¥27,800
- ⊙GF-730(730MB、10ms)…………特価¥37,800
- ⊙GF-1000(1060MB、9ms)………特価¥48,800

■ロジテック
- ⊙SHD-BA340U(340MB、12ms)……特価¥25,800
- ⊙SHD-BA540U(540MB、10.5ms)…特価¥28,800
- ⊙SHD-BA1000U(1GB)……………特価¥51,800

■システムサコム(富士通純正ドライブ使用)
- ⊙HD-M520(520MB、12ms)………特価¥37,800

内蔵

■CZ-500C/300C専用
- ⊙CZ-5H08(80MB/23ms)……定価¥ 98,000▶特価¥71,800
- ⊙CZ-5H16(160MB/18ms)……定価¥135,000▶特価¥99,500

●価格は変動します。ご注文の際は必ずお電話で価格と在庫をご確認下さい。●本広告に掲載の商品には送料及び消費税は含まれておりません。


ツクモイメージキャラクター
生田智子

The PlayStation BOOKS

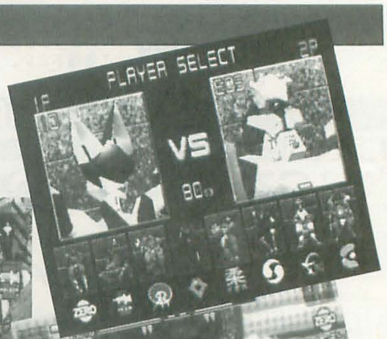
The PlayStation BOOKS

プレイステーションの定番ソフトを完全ガイド!!

中ボス・最終ボスの使用法まで完全ガイド！

ZERO DIVIDE
[ゼロディバイド]
パーフェクトガイド

10月上旬発売予定！







A5判・予価880円

プレイステーション用ロボット対戦格闘ゲーム「ZERO DIVIDE」を徹底攻略！プレイヤーキャラクター8体の対CPU・対人攻略を中心に、ボスキャラ・隠しキャラ「NECO」の操作法、隠しシューティングゲーム「Tinyファランクス」の紹介まで網羅した完全攻略本。「ZERO DIVIDE」をより一層楽しむための要素を満載した、ユーザー必携の一冊。　© ZOOM

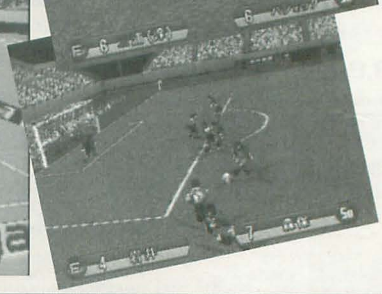
これさえあれば好きなクラブで優勝できる！

Jリーグサッカー
プライムゴールEX
パーフェクトガイド

10月上旬発売予定！







A5判・予価880円

Jリーグ公認のプレイステーション用サッカーゲーム「JリーグサッカープライムゴールEX」の完全攻略ガイド。全7種類にも及ぶモードの解説、基本テクニック、全クラブ選手データ、フォーメーション解説、高等テクニックを駆使した必勝パターン紹介など盛りだくさんな内容。この本があれば自分の好きなクラブで優勝できること間違いなし！　©（株）ナムコ

好評発売中！



究極のエースパイロットをめざせ！

エースコンバット
パーフェクトガイド
A5判・定価880円



アークザ ラッドを100%遊び尽くせる！

アーク ザ ラッド
パーフェクトガイド
A5判・定価880円



本誌で人気の格闘ライター杉澤教授が完全攻略！

鉄拳
パーフェクトガイド
A5判・定価880円



闘神伝の楽しさが、この一冊で倍増！

闘神伝
パーフェクトガイド
A5判・定価880円

●定価は税込みです　●お近くの書店でお求めください

SOFT BANK ソフトバンク株式会社/出版事業部
販売局　TEL. 03-5642-8101

原稿の、ファイルは雷雨とともに消え、ジャストのX68kペリフェラル

・・ですからねぇ、さっきから凄いんですよ。雷が。近所には落雷しまくるし、いきなり停電30分、打ち込んでいた原稿もその内容もすっかり忘却の彼方へ消え、停電が復旧したあともしばらく茫然自失状態に陥っておりました。しかも現在第二波接近中（しくしく）。で、つい先程まではいろいろなことを書いてましたが、一度書いてしまうとどーしても思い出せません。そうだ、免停処分通知あと1か月（推定）って話もありましたね。ただただ落ち込むだけです。あとは今週末晴海で開催のハムフェアに今年の夏のすべての情熱を賭けるだけです。も一夏なんて嫌いだぁ！、グレてやるー。・・あ、広告でしたっけ。はいはい。

▽拡張SIMM メモリーボード ER10S

型番：ER10S0n（SIMM未実装）定価￥14,800；ER10SDn（4MByte SIMM 1枚実装済）定価￥39,800　対応機種：X680x0 全機種（定価はすべて税別）

□クロックスピード20MHzオーバーのRISCチップを載せたプリンターがはびこる世の中、クロックスピード10MHzのX68000、今さらながら速い処理速度とは言えなくなりました。□68000の10MHZもさることながら、このクロックスピードに合わせたメモリー周辺の設計も足を引っ張る要因となっています。これではMPUのクロックを上げてもその効果が充分に生かされないこととなってしまいます。□H.A.R.Pの設計段階で判明していたMPUの高速化に伴うバス等でのウェイトタイム増大。この無駄な時間をより有効に活用するためのアーキテクチャーがER10の顔です。□H.A.R.P側から見た場合、MPU内部の倍速化された演算処理はストレートにバスに反映されるものの、メモリーアクセスに際しては既存クロックのサイクルで動作するバスのタイミングにあわせた動作をしなければならず、結果として常にウェイトが入っているような状態となります。□ここでER10をバスに接続した場合、バス側で4クロックをワンサイクルとするメモリーアクセスに対し、倍速動作のMPUクロックのアドバンテージを生かし、バス側で1クロック短縮した形でアクセスを完了できるようにタイミングを取る設計としています。□さらに、高速タイプの入手が容易な72ピンタイプのSIMMを採用、さらに内部で使用するゲートICなども高速のものを採用し、全体的な信頼性と安全性の向上に努めています。□「H.A.R.Pでない人」（笑）にもメリットがあると思いますよ。ER10、いかがですか。

▽MPUアクセラレーター H.A.R.P for MC68000

型番:DCMA00D1 定価￥29,800　対応機種:X68000初代,ACE,EXPERT,SUPER

マシンは速くしたい、改造は自信がない、費用も押さえたい。三拍子そろったあなたの欲求、H.A.R.Pがまとめてお引き受けいたします。□既存のMPUと交換するだけであっという間に倍クロック動作、周辺回路とのタイミングはクロックアップ前の状態を保ったまま、電気的に負担をかけることなく手軽に高速化。ソフトウェア的な互換性をパッチ不要のまま高いレベルで実現しています。□さらに拡張メモリーボードER10Sと組み合わせられることにより、メモリーアクセスのボトルネックを改善、トータルで約50％（弊社測定値）のパフォーマンスアップが可能です。H.A.R.Pの性能を確実に引き出すには両者を組み合わせて使うのがベストですょ。□手軽なインストレーションと優れたコストパフォーマンス、H.A.R.Pは常にあなたの強い味方ですょ。

▽拡張 I/O スロット ESX68

型番：ESX68L4 予価￥39,800 対応機種：X680x0 全機種

OS-9をはじめ、実はFA系での隠れた需要もあるX680x0、この辺の用途にご利用の皆様には特に拡張 I/O スロットの少なさが問題となっているかと思います。□そんな需要家の皆様、そして純粋にコンピューティングを楽しむユーザーの皆様、外部拡張 I/Oスロットはいかがでしょうか？□本体電源に連動する外部スロット専用電源を内臓し、X68k本体とのインターフェースカードは高速タイプのバッファを搭載。加えて3スロットが追加利用できます。□LAN,PIO,GPIB, 入れたいカードは何でもどうぞ。□結構シリアスな設計しました。ESX68、くどいようですが、よろしくどーぞ。

▽MPUアクセラレーター H.A.R.P-FX（H.A.R.P for MC68030）

型番：DCMA30F1　予価￥54,000
対応機種：X68030をはじめ、MC68030（PGAソケット）が採用されたコンピュータシステム（供給クロック25MHZ以下）

□X68030をはじめPGAパッケージタイプ68030を採用するパーソナルコンピューター、ワークステーションのほとんどに適応可能なMC68030互換MPUアクセラレーター、H.A.R.P-FXですX68030への実装時には 25MHzのクロックを2倍、オンボード上のMC68030RC50へフルスペック50MHzクロックを供給し、さらにMPUオンチップのキャッシュメモリーがクロックスピードと相乗し優れたパフォーマンスを発揮してくます。もちろん、ソフトウェアの互換性を完全に維持、既存の環境で動作していたソフトウェアならまず問題なく実行可能でしょう。Pentiumの120MHzもいいですが、68030の深い味わいを放つH.A.R.P-FX、ひたすら我が道を突き進みます。ご期待ください。

・・ちょっとグレてみました（担当）。

サポート　（有）エヌ・エム・アイ

開発・販売　（株）ジャスト
〒156　東京都世田谷区宮坂3-10-7 YMTビル3F
Phone.03-3706-9766　FAX.03-3706-9761　BBS.03-3706-7134

JAST

GAME BEST SELECTION
ゲームベストセレクションシリーズ

ソフトバンク株式会社
出版事業部　SOFT BANK

●定価は税込です　●お近くの書店でお求めください

米国「Codies賞」受賞！
超話題の純国産シミュレーションソフトを完全攻略!!

絶賛発売中



Tower［タワー］
公式パーフェクトガイド

◎グレードを上げるための数々の条件をクリアし、思い通りのビルを建築する様々なテクニックを徹底解説。
◎秘密の裏ワザ、コマンドなども完全紹介。
◎困ったときにすぐ役立つ〈INDEX〉付き。

山猫有限会社 著

昨年発売された中で最も優れたソフトに与えられる権威ある「Codies賞」を受賞した、大ヒット純国産シミュレーションゲーム「Tower」公式完全ガイド。最高グレードである〈Tower〉の称号をもらうまでの様々なテクニック、自分の好きなビルを建築するためのノウハウなど、「Tower」のすべてを徹底解説！

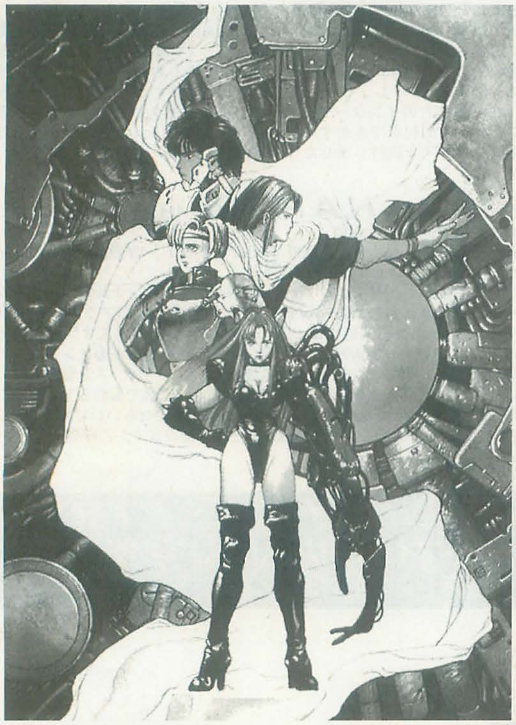
A5判・定価1,600円

SOFTBANK GAME BOOK SELECTION

SOFTBANK CD･BOOK

メモリアルドラマCD＆ファンブック

ファンタシースター



9月下旬発売予定!

セガのゲームマシン（マークIII、メガドライブ）を代表するファンタジーRPG「ファンタシースター」ファンブック。セガ監修によるオリジナルドラマCD付き。ドラマCDは、シリーズ4作目「千年紀の終りに」の世界をベースに、シリーズで人気の高いキャラクター、ネイを使ったオリジナルシナリオによるものです。出演している声優のインタビューなども収録。

B5判・定価3,900円

人気声優が出演!

ネイ……………三石琴乃
スレイ…………速水奨
フォーレン ……井上和彦
ルディ …………阪口大助

© SEGA

ファンブックと揃えて、
コレクターズアイテムに加えよう!

ファンタシースター 公式設定資料集

「ファンタシースター」の公式設定資料集。これまでに発売されたIから「千年紀の終りに」までの未公開資料を含め、すべての資料を網羅。アルゴル太陽系年表完全版や用語集、開発者スペシャルインタビューなどから、ファンタシースター通販グッズ、すべてのテレビCMまで大紹介。

綴じ込み付録
米田仁士氏
描き下ろし
B3判
特製ポスター

B5判・予価1,500円

10月上旬発売予定!



SOFTBANK CD･BOOK

メモリアルドラマCD&ファンブック

ラングリッサーII

好評発売中!

秘剣“ラングリッサー”をめぐって、激しい戦いの幕が開こうとしている。エルウィンとレオンとの宿命の戦いが、いま始まる!!そしてファンブックでは、うるし原智志氏デザインによるキャラクターの魅力を徹底紹介&「ラングリッサー」シリーズの歴史を検証。そのほか、特別インタビューやメイキングなどを収録。

B5判・定価3,800円

© NCS

豪華声優陣を起用!

【CAST】

エルウィン …草尾　毅
シェリー ……横山智佐
リアナ ………國府田マリ子
エリザ ………林原めぐみ
エグベルト …青野　武
ナレーター …銀河万丈
ジェシカ……滝　恵子
ヘイン………山口勝平
レオン………置鮎龍太郎
レアード……堀川　亮
バルガス……郷里大輔

●定価は税込みです　●お近くの書店でお求め下さい

ソフトバンク株式会社／出版事業部 SOFT BANK
販売局 TEL.03-5642-8101



セクシーでパワフルな女子プロを制覇しろ！

18禁版

カードバトルにプロレスを融合させた、「レッスルエンジェルス」シリーズ。いよいよ最大のヒット作「レッスルエンジェルススペシャル」が登場です。さまざまなイベントの選択によって運命が変わる、マルチシナリオ・マルチエンディング。プロレス技数、カテゴリーが増加して、レスラーの個性もパワーアップ。そして、「恐怖の水着はぎデスマッチ」もパワーアップして復活！18禁だから、そのセクシー度はもうケタ違い！待望のX68000移植完成！明日のトップイベンターを目指すのだ！



機能アップ！

- ●オリジナルオープニングを収録
- ●画面のレイアウトを変更
- ●エキジビションモードグラフィック描き直し
- ●256色モードと16色モードを搭載
- ●サウンドも明るめに変更
- ●AD-PCMによる効果音
- ●ディスクアクセスを最少に抑える設計

このソフトは、全国のパソコンショップで、パッケージ版で販売いたします。TAKERUでは販売致しません。TAKERU事務局では通信販売はいたしませんので、悪しからずご了承下さい。

対応機種：X68000/X68030
要メモリ2Mバイト
（ハードディスク対応）
制作：グレイト

￥8,800（税別）



三國志

知力の極限に挑む、君主、武将、軍師の膨大なデータ。小説よりリアルと、名作の誉れ高い中国統一ゲーム。この歴史的な傑作シリーズはどのようにして始まったのか?SLGファンなら絶対に見逃せない!

制作/光栄
対応機種/X68000（30不可） **¥5,200**



三國志 Ⅱ

登場人物350余名、最大11人まで同時プレイ可能、6編のマルチシナリオ方式、埋伏の毒・駆虎呑狼等のユニークな計略要素導入、さらに深みを増した外交・HEX戦など、まさに名作!カシオペアの向谷　実のBGMも話題に。

制作/光栄
対応機種/X68000（30不可） **¥4,900**



大航海時代

リコエイションゲームシリーズの傑作。毎回違った展開が楽しめるイベントジェネレーティングシステム。帆船の特徴が活かされたHEX戦。失われたロマンを求めて、冒険者たちの航海の旅が始まる。

制作/光栄
対応機種/X68000（30可） **¥3,400**



維新の嵐

坂本龍馬が、西郷隆盛が、吉田松蔭が日本を憂い、改革を目指して奮い立つ!幕末の志士の個性を際だたせる緻密なパラメータ。出会いの楽しさ、駆け引きを楽しむ新システム。強力な機能で、維新を操れ!

制作/光栄
対応機種/X68000（30不可） **¥3,400**



信長の野望　戦国群雄伝

400余名の群雄が割拠する下剋上の乱世。配下の羽柴秀吉、柴田勝家を個性豊かな武将たちを、思いのままに操って、戦雲たなびく戦場へ、天下分け目の決戦に臨む！光栄の代表作「信長の野望」シリーズの傑作！

制作/光栄
対応機種/X68000（30可） **¥3,400**



伊忍道　打倒信長

1つのゲームでSLG とRPG、2つのジャンルが楽しめるリコエイションゲームの第3弾。特にRPGの要素が濃い、異色傑作だ!意志を持ったキャラクターが目的に向かって行動を展開。敵を倒して腕を上げ、技を磨いて信長を倒せ!

制作/光栄
対応機種/X68000（30不可） **¥3,400**



太閤立志伝

裸一貫の足軽頭から身を興し、関白にまで登り詰めた男・木下藤吉郎(豊臣秀吉)。草履を温めたエピソード・奇跡の墨俣一夜城など、数々の逸話を持つ男の一生を再現する、リコエイションゲームの傑作です。

制作/光栄
対応機種/X68000（30不可） **¥3,400**



蒼き狼と白き牝鹿　元朝秘史

光栄歴史三部作の一角を成す、草原の英雄チンギス・ハーン。稀代のスケールと空前絶後の迫力で、一代帝国を築き上げた男の豪快な一生を見事に再現!熱いシミュレーションの傑作です。

制作/光栄
対応機種/X68000（30不可） **¥3,400**



ロイヤルブラッド

新シリーズ「イマジネイションゲーム」のデビュー作。イシュメリアという架空の島国を舞台にした、幻想世界のシミュレーションゲームだ。あなたは独立貴族のひとりとなり、領主達が持っている6つの宝石を集め、イシュメリアの新王となれ!

制作/光栄
対応機種/X68000（30可） **¥2,700**



ヨーロッパ戦線

戦乱のヨーロッパ。砂塵の彼方から迫り来る黒い車体は、敵か味方か?次々に飛び込んでくる情報、時事刻々と変わる戦局。多彩な兵器やユニット、人間的要素を重視した各種パラメータ。WWIIシリーズ第2弾。勝利の旗を手に入れろ!

制作/光栄
対応機種/X68000（30可） **¥4,500**



大戦略 Ⅲ '90

90年代にふさわしくパワーアップされた「大戦略」シリーズ。戦略思考ルーチン、ゲームスピード、コマンド体系、リアルタイムオペレーションなど大幅革新された作品です。

制作／システムソフト
対応機種／ X68000 **¥2,500**



ジェノサイド 2

あのズームのゲームがついに名作文庫に登場！特大キャラとハデハデな演出で、68ユーザーのどぎもを抜いた名作アクションゲームだ。MIDIにも対応しているぞ。

制作/ズーム
対応機種/X68000（30不可） **¥2,500**



ファランクス

デカキャラ・派手め演出の横スクロールオアワーシューティング。拡大・回転・縮小・多関節・半透明・ラスタースクロール・MIDIと、各種要素がいっぱい詰まってます。

制作／ズーム
対応機種／X68000（30不可） **¥2,500**



A列車で行こうⅡ

かの「A列車」シリーズの第2弾。パズル的要素がアツクなる!鉄道会社社長の立場で、線路の敷設・撤去を行い、ワールドワイドにマップを発展させていこう。

制作／アートディンク
対応機種／X68000（30不可） **¥3,800**



A Ⅲ（A列車で行こう3）

さらにワイドに、さらに完成度の増した、世界レベルヒットの第3弾。世にA.Ⅲブームを巻き起こしたことで、記憶に新しい超有名作、ついに文庫に登場！

制作／アートディンク
対応機種／X68000（30可） **¥3,800**



栄冠は君に

高校野球シミュレーションシリーズの、記念すべき第1作。全国制覇を達成するには、3990校の頂点に立たなければならない。感動の優勝セレモニーを、果たして見ることが出来るか！？

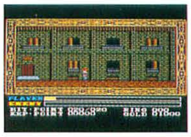
制作／アートディンク
対応機種／X68000 **¥3,800**



ルーンワース「黒衣の貴公子」

ハイドライドシリーズに続く、新ARPGシリーズ第1弾。綿密に構築された世界「ルーンワース」を舞台に、極めて自由度の高いゲームシステムの中で、興奮の冒険が始まります。

制作／T&Eソフト
対応機種／X68000 **¥ 700**



イース Ⅲ（ワンダラーズフロムイース）

よりアクション性を増した、これまた、大人気を博したアクション・ロールプレイング。アドルの最後の冒険物語でした。攻撃方法もいっそう多彩になって、時間を感じさせない逸品です。

制作／日本ファルコム
対応機種／X68000（30不可） **¥2,000**

パソコンソフト
自動販売機
TAKERU

TAKERU事務局
〒467 名古屋市瑞穂区苗代町2番1号
ブラザー技術開発センタービル2Ｆ
TEL(052)824-2493（受付時間：月～金　13:00～18:00）

営業所
東京営業所
(03) 5443-4967
大阪営業所
(06) 258-3024

通信販売 1994年4月1日より、送料／手数料が有料になりました。

ソフト名、機種名、メディアのサイズ、住所、氏名、電話番号を明記の上TAKERU事務局まで現金書留でお申し込みください。送料／手数料は、1回のお申し込み総金額が5,000円以上の方は無料。4,900円までの方は500円をいただきます。4,900円までの方は現金500円をプラスしてお申し込みください。誠に勝手ながら、皆様のご理解とご協力の程、お願い申し上げます。

SHARP

目の付けどころが、シャープでしょ。

感性を光らせる。

さまざまなフィールドで、研ぎ澄まされた感性に応える潜在能力の実証

X68の潜在能力は、まさに時代とともに証明されつつあります。
開発当初より、現在のマルチメディア環境を想定していた事実。
グラフィック能力はもちろん、ADPCM対応、オリジナルウィンドウシステム、
X68にとってこれらは、数年前のスペックなのです。
パソコンの存在そのものを革新した「創造性」、マインドを喚起する「こだわり」、
いま、先見のユーザーに支えられたX68は
そのコンセプトの開花を得て、多彩なフィールドへと飛翔します。

Workbench

WSとしての楽しみ

たとえば、リアルタイム・マルチタスク・オペレーティング・システムOS/9。
X68030の能力を最大限に引き出す
UNIXライクな操作性と洗練された機能。
X-WINDOWや動画ツールのサポートで
さらに深い楽しみが…。

※OS/9はマイクロウェア・システムズ㈱の登録商標です。
※UNIXは、X/Openカンパニーリミテッドが独占的にライセンスする米国および他の国における登録商標です。

Create

創造するよろこび

SX-WINDOW開発支援ツールが
創造力を刺激する。
ソフト開発に必要なツールや
サンプルプログラムを多彩にバンドル、
ウィンドウ上で効率よく作業でき、
初めてプログラムに挑む人への
やさしい配慮が、創造するよろこびを
さらに高めてくれるでしょう。

Ammusement

遊びへのこだわり

X68の能力の高さを端的に示す
アミューズメントフィールド。
マインドをきわめたゲームフリークの
熱い期待に応える。
画像の美しさが感性を刺激する、
さらにパワーアップされた
「スーパーストリートファイターII」なら、
キミのこだわり度は今、全開！

© CAPCOM ALL RIGHTS RESERVED



X68030 32bit PERSONAL WORKSTATION / **X68000** PERSONAL WORKSTATION ·XVI

X68030 [本体+キーボード+マウス・トラックボール]
130㎜FD(5.25型)タイプ CZ-500C-B(チタンブラック) 標準価格398,000円(税別)・〈HD内蔵〉CZ-510C-B(チタンブラック) 標準価格488,000円(税別)

X68030 Compact [本体+キーボード+マウス]
90㎜FD(3.5型)タイプ CZ-300C-B(チタンブラック) 標準価格388,000円(税別)

X68000 XVI Compact [本体+キーボード+マウス]
90㎜FD(3.5型)タイプ CZ-674C-H(グレー) *

●ディスプレイは別売です。●消費税及び配送・設置・付帯工事費、使用済み商品の引き取り費等は、標準価格には含まれておりません。●画面はハメコミ合成です。
*〈標準価格〉表示のない商品の価格については、販売店にお問い合わせください。

■お問い合わせは… シャープ株式会社 電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)

T1002179100764 雑誌 02179-10